The Thesis Committee for Wenjun Li
Certifies that this is the approved version of the following thesis:

Linear Solvers and Coupling Methods for Compositional Reservoir Simulators

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Kamy Sepehrnoori

Mojdeh Delshad

Linear Solvers and Coupling Methods for Compositional Reservoir Simulators

by

Wenjun Li, B.S.,M.S.,Ph.D.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2010

Dedication

To

my wife, Jenny Sun,

and my lovely son, Tony Li,

for their unconditional love and support

Acknowledgments

I have been lucky to have the privilege of studying at The University of Texas at Austin. During my studies at UT, many people helped me in different ways in completing this work.

First, I would like to take this opportunity to express my sincere gratitude to Professor Kamy Sepehrnoori, my advisor, for his invaluable guidance, warm encouragement for this study. I am deeply indebted to him for his advice, his patience, and his expertise, his confidence in my research ability. His abundant knowledge in petroleum and computational mathematics and his skillful ways of analyzing and solving problems supplied me not only guidance to my research, but also the true scientific and engineering methodology in order to handle the real world of petroleum engineering. Special thanks go to Professor Mojdeh Delshad for her guidance for this thesis, her help on the HYPRE solver for UTCHEM during the development phase, and her helpful training class of UTCHEM.

I am highly grateful for Professors Mary Wheeler and Todd Arbogast for their fruitful discussions about the mathematical issues on the preconditioner for the partition method. Also I would like extend special thanks to Dr. Lee Chin, who has been a collaborator of my research on the partition method. I also would like thank Dr. Mohammad Hosein Kalaei for proofreading my thesis in his valuable time and his helpful suggestions for improving it.

I also would like to express my appreciation to Professor Klaus Stüben for his permission to use SAMG for our reservoir simulators, and his great help on selecting

optimal parameters of SAMG for the tests in this thesis. The same appreciation goes Dr. Robert D. Falgout for his great help on compiling and linking HYPRE under Windows OS. I also appreciate Mr. Steve Robinson at Tecplot Inc. for supplying me a license for Tecplot 360 2010, and his great help on how to apply Tecplot in this thesis.

It has been a wonderful experience to be part of the Research Simulation Joint Industry Project, and I wish to thank our group members for providing me invaluable assistance in all aspects of my research. Also I would like to express my special thanks to Cheryle Kruzie, Dr. Roger Terzian, Tim Guinn, Mary C. Pettengill, and Joanna Castillo for their great help, and I would like to extend my many special thanks to my Chinese friends in the Department of Petroleum and Geosystems Engineering for sharing their friendship with me.

I greatly acknowledge the financial supports of a RA (research assistantship) from the Center of Petroleum and Geosystems Engineering through Professor Sepehrnoori and a TA (teaching assistantship) from the Department of Petroleum and Geosystems Engineering through Professor Sepehrnoori at The University of Texas at Austin (UT-Austin) toward my master science program. Also, I would like to deeply acknowledge the Cockrell School of Engineering School at UT-Austin for the kind support and sincere encouragement of both the EPJLT Annual Scholarship and the CLAW Annual Scholarship during my study at UT-Austin.

Lastly, I would like to express my deepest gratitude to my wife, Jenny Sun, and my son, Tony Li for their love, patience, understanding, encouragement, and continuous support which supplied me the source of inspiration to finish this study.

Abstract

Linear Solvers and Coupling Methods for Compositional Reservoir Simulators

Wenjun Li, M.S.E.

The University of Texas at Austin, 2010

Supervisor: Kamy Sepehrnoori

Three compositional reservoir simulators have been developed in the Department of Petroleum and Geosystems Engineering at The University of Texas at Austin (UT-Austin): UTCOMP (miscible gas flooding simulator), UTCHEM (chemical flooding simulator), and GPAS (General Purpose Adaptive Simulator). UTCOMP and UTCHEM simulators have been used by various oil companies for solving a variety of field problems. The efficiency and accuracy of each simulator becomes critically important when they are used to solve field problems. In this study, two well-developed solver packages, SAMG and HYPRE, along with existing solvers were compared. Our numerical results showed that SAMG can be an excellent solver for the usage in the three simulators for solving problems with a high accuracy requirement and long simulation times, and BoomerAMG in HYPRE package can also be a good solver for application in the UTCHEM simulator.

In order to investigate the flexibility and the efficiency of a partitioned coupling method, the second part of this thesis presents a new implementation using a partition method for a thermal module in an equation-of-state (EOS) compositional simulator, the General Purpose Adaptive Simulator (GPAS) developed at The University of Texas at Austin. The finite difference method (FDM) was used for the solution of governing partial differential equations. Specifically, the new coupled implementation was based on the Schur complement method. For the partition method, two suitable acceleration techniques were constructed. One technique was the optimized choice of preconditioner for the Schur complement; the other was the optimized selection of tolerances for the two solution steps. To validate the implementation, we present simulation examples of hot water injection in an oil reservoir. The numerical comparison between the new implementation and the traditional, fully implicit method showed that the partition method is not only more flexible, but also faster than the classical, fully implicit method for the same test problems without sacrificing accuracy. In conclusion, the new implementation of the partition method is a more flexible and more efficient method for coupling a new module into an existing simulator than the classical, fully implicit method.

The third part of this thesis presents another type of coupling method, iterative coupling methods, which has been implemented into GPAS with thermal module, FICM (Fully, Iterative Coupling Method) and GICM (General, Iterative Coupling Method), LICM (Loose, Iterative Coupling Method). The results show that LICM is divergent, and GICM and FICM can work normally. GICM is the fastest among the compared methods, and FICM has a similar efficiency as CFIM (Classic Fully Implicit Method). Although GICM is the fastest method, GICM is less accurate than FICM for in the test cases carried out in this study.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background and Literature Review

This section describes three compositional reservoir simulators that we used in this thesis, AMG method and its software packages that were added into the three simulators, coupling method, and thermal module of GPAS.

1.1.1 Three Compositional Reservoir Simulators

The Department of Petroleum and Geosystems Engineering (PGE) at The University of Texas at Austin (UT-Austin) has three well-known compositional reservoir simulators called The University of Texas at Austin Compositional simulator (UTCOMP), The University of Texas at Austin Chemical Flooding simulator (UTCHEM), and General Purpose Adaptive Simulator (GPAS).

UTCOMP is an isothermal, three-dimensional, compositional simulator for miscible gas flooding. The solution scheme used in this simulator is Implicit Pressure Explicit Concentration (IMPEC), which is analogous to Implicit Pressure Explicit Saturation (IMPES) (Chang, 1990), in which the grid block pressure is solved for implicitly using explicit updating of saturation-dependent terms first, then the component mass balance equations are solved explicitly for the overall molar concentration of each component. UTCOMP can simulate the compositional reservoir flow featured with up to four phase compositions such as aqueous phase, oil phase, gas phase, and a second non-aqueous liquid phase obtained by flash calculations based on a rigorous Gibbs stability test and a phase identification test, the non-aqueous fluid properties modeled using the Peng-Robinson equation-of-state (PRES) or a modified version of the Redlich-Kwong equation-of-state (RKES).

UTCHEM is the three-dimensional multiphase multi-component chemical compositional simulator (Delshad et al. 1996). UTCHEM can handle very complex chemical flooding simulations with a comprehensive surfactant/oil/brine phase behavior, petrophysical properties, chemical reaction, and heterogeneous porous media properties, and four fluid phases (gas, aqueous, oleic, and microemulsion) and solid phases. .

GPAS is a next generation compositional reservoir simulator after UTCOMP and UTCHEM simulators, which has been under development by PGE Department at UT-Austin since 2001. GPAS is a fully implicit, compositional, equation-of-state, parallel simulator that can run both a single PC and massively parallel computers or clusters of PCs, which contains many modules such as compositional, chemical flooding, thermal, naturally fractured reservoir, geomechanics, asphaltene precipitation and wellbore modules (John et al., 2004; Han et al., 2005; Naimi-Tajdar et al., 2006; Pan et al., 2009; Tarahhom et al., 2009; Fazelipour et al., 2008; and Varavei et al., 2009).

1.1.2 Algebraic Multigrid (AMG) and Its Software Packages

The multigrid principle, which requires hierarchical algorithms which ensure a rapid reduction of both short and long range error components, has made possible the efficient numerical solution of large systems of discretized elliptic partial differential equations (PDEs) during the last three decades. "Geometric" multigrid is a method that operates on a hierarchy of grids, defined a priori by coarsening the given discretization grid in a geometrically natural way. The hierarchy of grids is straightforward for logically regular grids; it becomes very complicated for highly complex, unstructured meshes.

The automatic coarsening process took place in the early eighties (Brandt et al., 1982, 1984, 1986). At the time, it was based on the combination of the so-called Galerkin-principle and operator-dependent interpolation in geometric multigrid to

increase its robustness aiming at the efficient solution of diffusion problems with jumping coefficients (Dendy, 1982; Alcouffe, 1981). The result was a multigrid-like approach which did not merely allow an automatic coarsening process, but could directly be applied to (linear sparse) algebraic equations of certain types, without any pre-defined hierarchy ("algebraic" multigrid, AMG).

The first, fairly general AMG program was described and investigated by Ruge et al. (1985, 1986) and Stüben (1983). The code AMG1R5 was made publicly available in the mid-eighties. However, since the early and mid-nineties, the interest in algebraically oriented multilevel methods increased strongly due to the limitation of geometric mulgrid, and the increase demand for the efficient "plug-in" solvers.

One such AMG software package is SAMG (Algebraic Multigrid Methods for Systems), which was written in FORTRAN90 based on an AMG (Algebraic Multigrid) approach, is a library of subroutines for the highly efficient solution of large linear systems of equations with sparse matrices (Stüben and Clees 2005). Such systems of equations form the numerical kernel of most simulation software packages. Usually, the numerical solution of these linear systems of equations requires most of the computational expense of the whole simulation for classical methods. However, SAMG has the advantage of being almost unconditionally numerically scalable. This means that the computational cost using SAMG depends linearly on the number of unknowns.

Essential components of SAMG have been implemented already in the code RAMG (Stüben 2001), which is a successor of the original code AMG1R5, described in Ruge and Stüben (1986). Compared to RAMG, however, SAMG is much more general. In particular, SAMG can be applied to both scalar and coupled systems of elliptic PDEs. SAMG is a "plug-in" solver. Essentially, just the (sparse) linear system of equations to be solved, Ax=b, has to be passed to SAMG. In general, no information regarding the shape

of the domain or the structure of the underlying grid needs to be provided. Thus, besides its robustness and efficiency, the main practical advantage of SAMG is that it can directly be applied to solve certain classes of PDEs on unstructured meshes, both in 2D and 3D. Moreover, SAMG can even be applied to problems without any geometric background information, provided that the underlying matrices are of a similar type as the ones arising from elliptic PDEs.

Another AMG software package is HYPRE. HYPRE is a software library of high performance preconditioners and solvers for the solution of large, sparse linear systems of equations on massively parallel computers (CASA 2008). The HYPRE library was created with the primary goal of providing users with advanced parallel preconditioners. The library features a parallel multigrid solver, BoomerAMG, for both structured and unstructured grid problems. BoomerAMG is a parallel implementation of the algebraic multigrid method (Ruges and Stüben 1987). It can be used either as a solver or as a preconditioner. The user can choose between various different coarsening techniques, interpolation, and relaxation schemes. See Henson and Yang (2002), Yang (2006) for a detailed description of the coarsening algorithms, interpolation, and relaxation schemes as well as numerical results.

1.1.3 Coupling Methods

Coupling methods became gradually popular when new simulation modules needed to be added to existing software packages. This thesis focuses on two kinds of coupling methods such as two iterative coupling methods and a partition method. Settari and Mourits (1994) first proposed and applied the iterative coupling method for the coupling between the geomechanics model and the reservoir simulation model (Settari and Mouris 1994). Then the iterative coupling methods were improved and applied for more

complicated reservoir models (Settari et al. 1995, 1999, and 2000). In 2002, a more general implementation of the iterative coupling method, which did not need to modify original reservoir simulator's source code, was proposed and applied to some general reservoir models (Chin et al. 2002). Gai et al. (2003, 2004), and Dean et al. (2006) performed the coupling geomechanics procedure with a black-oil model by following Settari et al.'s strategy (Gai et al. 2003, Gai 2004, and Dean et al. 2006). Pan implemented the iterative coupling method using Chin's algorithm for the geomechanics model and GPAS (Pan et al. 2007). Settari et al. 1995 performed further research on the iteratively coupled process (Settari et al., 1995, 1997 and 1999; Tran et al., 2004a, 2004b, 2005). Lu et al., (2007) applied the iteratively coupled reservoir simulation for multiphase flow.

Prevost proposed a fully coupled method, the partitioned method (Prevost, 1997). Prevost solved a linear system of partial differential equations discretized using a finite element method that was coupled with a new physical model. Since the linear system was a symmetric positive definite (SPD), a preconditioned conjugate gradient (PCG) method was employed in the two-step solving procedure to obtain the original unknowns and the new coupled unknowns synchronously without assembling the coupling matrices and the related Schur complement matrix.

1.1.4 Thermal Modules

The thermal model simulation based on equation-of-state (EOS) has been studied for more than two decades. Ishimoto first developed a one-dimensional and fully implicit compositional steam-flood model (Ishimoto 1985, Ishimoto et al. 1987). The fully implicit, four phase multi-component, multidimentional steam and combustion simulator, which included a fully implicit well model and had appropriate and robust iterative

techniques for solving large thermal problems, was presented by Rubin and Buchanan in 1985. Chein et al. (1989) developed a general purpose compositional simulator in which the thermal model selects K-values or the EOS to calculate fluid properties. Trangenstein 1989 analyzed a two-component, three-phase flow thermal model. Considering water as a non-ideal component and enthalpy as a primary variable, Brantferger et al. (1991) developed a novel simulator with improved formulation.

There are different approaches to implement EOS-based thermal compositional simulators. A fully coupled, fully implicit simulator, in which its frame-work supports an IMPES and a sequential semi-implicit formulation, was developed by Miffin et al. in 1991. Chan and Sarioglu described a procedure for incorporating fracture characteristics in a thermal reservoir simulator in 1992. Cicek et al. (1996) developed a compositional, fully-implicit 3D simulator that can handle steam injection. In 2005, Cicek also improved his simulator so that it can handle naturally fractured reservoirs. He tested the steam displacement of oil in a naturally fractured reservoir and investigated a two-phase liquid region by a steam condensation front. The thermodynamic properties and compositional effects of the phase behavior calculation are more accurate in Cicek's approach than those produced by previous simulators because the physical properties of phase and components were calculated based on EOS, and not on any steam-table form.

1.2 Research Objective

The work, performed in this thesis, concerns the efficiency improvement for three compositional simulators such as UTCOMP, UTCHEM, and GPAS. This thesis has two parts that resulted from two main research objectives.

The first part describes the investigation of the possible improved efficiency by implementing the new SAMG and BoomerAMG solver as options into UTCOMP, UTCHEM, and GPAS based on the available linear solver packages such as SAMG and HYPRE and to compare them with the original linear solvers in the three reservoir simulators. Then the suitable options of linear solvers for usage in the three compositional simulators can be determined. The second part of this thesis is the coupling of a thermal module into GPAS by several coupling approaches such as a partition method and iterative coupling methods. The main purpose of the second part is to investigate the possibility of improving the efficiency and flexibility of adding a new thermal module into GPAS and to compare with the traditional implementation of fully implicit method.

In order to realize the above tasks, the objectives of this thesis are stated as follows:

1. Installation of SAMG software package and HYPRE software package on our server machine Petros.
2. Implementation of SAMG and BoomerAMG into UTCOMP.
3. Implementation of SAMG and BoomerAMG into UTCHEM.
4. Implementation of SAMG into GPAS.
5. Modification of a thermal module from an energy model equation in GPAS.
6. Development and implementation of a partition method and two iterative coupling methods for both thermal module and isothermal GPAS simulator.
7. Verifying the efficiency of the new methods mentioned above by numerical tests.

1.3 Brief Description of Chapters

Chapter 2 will briefly introduce the three compositional simulators UTCOMP, UTCHEM and GPAS. Next, linear solver packages of HYPRE and SAMG are introduced, and the related compilation, link, and installation of their libraries are presented.

The existing linear solver in UTCOMP and the implementation of SAMG and BoomerAMG in the UTCOMP and their numerical comparisons for the test cases are presented in Chapter 3. First, the UTCOMP mathematical model equations in PDEs (Partial Differential Equations) format are introduced. Next, the numerical discretization of the PDEs model in the FDM is presented. Then the IMPEC (Implicit Pressure Explicit Concentration) of solution procedure is described. The existing linear solver is also described. The implementation of SAMG in UTCOMP is detailed. In the last part, the results of several test cases are presented, analyzed, and discussed.

Chapter 4 follows the similar procedure as Chapter 3 to present the existing linear solver in UTCHEM and the implementation of SAMG and BoomerAMG in UTCHEM and their numerical comparisons for the test cases.

Chapter 5 is devoted to a similar procedure as described in Chapter 3 to present the existing linear solver in GPAS and the implementation of SAMG in the GPAS. The results of some cases are analyzed.

Chapter 6 describes the thermal module of GPAS, and the development and implementation of a partition method for a thermal module and isothermal GPAS.

Chapter 7 deals with the implementation of iterative coupling approaches, i.e., two iterative coupling methods, for the thermal module and isothermal GPAS, and the numerical results are presented to verify their efficiency and accuracy.

Chapter 8 presents a summary and the conclusions of this thesis, as well as recommendations for future work in developing more efficient approaches to increase both efficiency and flexibility of multiple models coupled reservoir simulator.

Chapter 2: Reservoir Simulators and Linear Solvers

2.1 UTCOMP Simulator

UTCOMP is an isothermal, three-dimensional, compositional simulator for miscible gas flooding which uses an IMPEC solution scheme for solving the governing partial differential equation. UTCOMP employs both Equation-of-State (EOS) and Black Oil Model (BOM) for the phase behavior of the reservoir fluids. UTCOMP is a serial software package written in the FORTRAN language. The current version of UTCOMP simulator used in thesis is Version 3.8, in which the source code includes 231 files. The current option of compiler is Intel FORTRAN. UTCOMP can be compiled, linked and run under both Windows and Linux/Unix operating systems. The flowchart of UTCOMP is presented in Figure 2.1-1, where

WIP: Water in place

HCIP: Hydrocarbon in place

OOIP: Origin oil in place

OGIP: Original gas in place

BOM: Look-up table form of phase behavior black oil model

EOS: Equation-of-State of phase behavior

BHP: Bottomhole pressure

GOR: Gas oil ratio

WOR: Water oil ratio

HC: Hydrocarbon

MIP: Materials in place

M: Materials in M produced and M injected

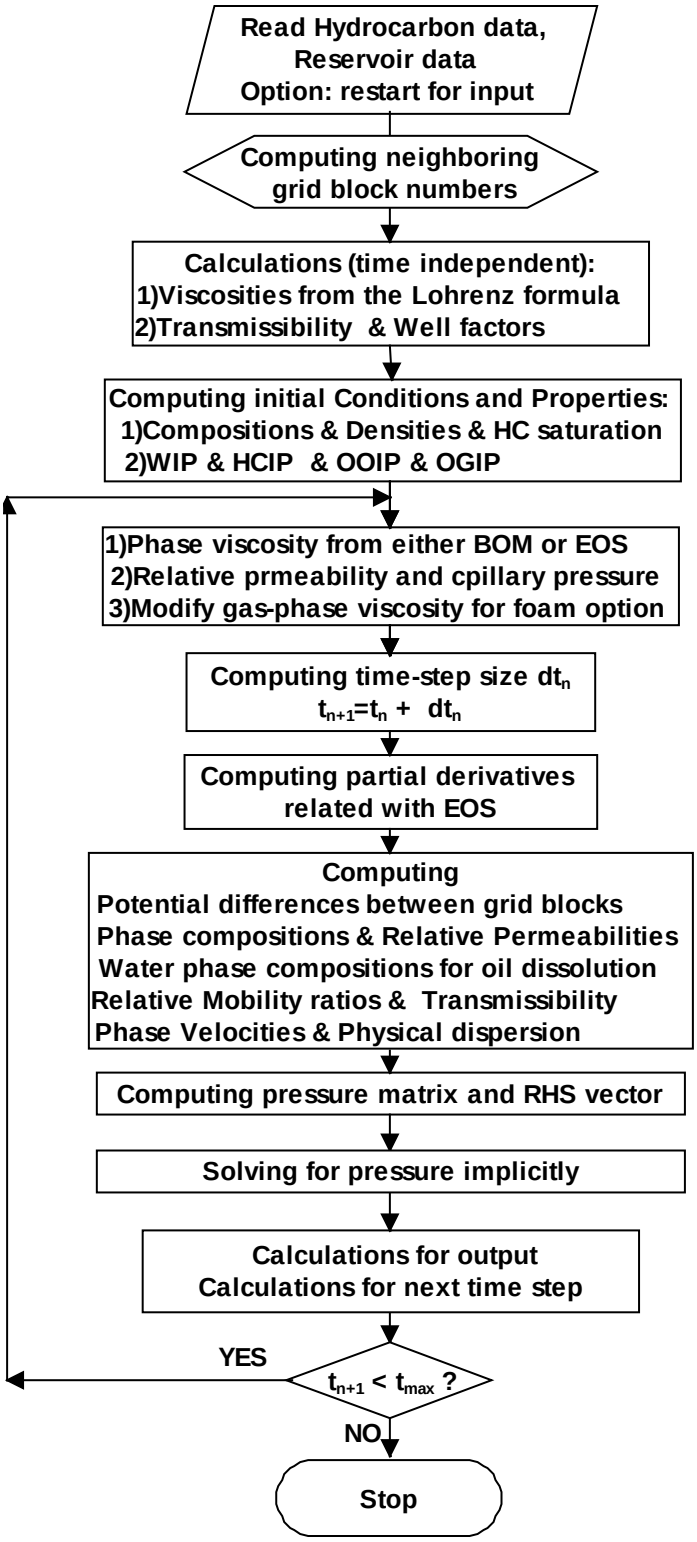

Figure 2.1-1: Flowchart of UTCOMP simulator.

2.2 UTCHEM Simulator

UTCHEM is a three-dimensional, multiphase, multi-component chemical compositional simulator. UTCHEM is a serial software package written in FORTRAN computer language featured by dynamic memory allocation. The current version of UTCHEM simulator used in this thesis is UTCHEM-D-9.9-2008, whose source code includes 115 files. The current compiler option is Intel FORTRAN. UTCHEM can be compiled, linked, and run under both Windows and Linux/Unix operating systems. The flowchart of UTCHEM is presented in Figure 2.2-1.

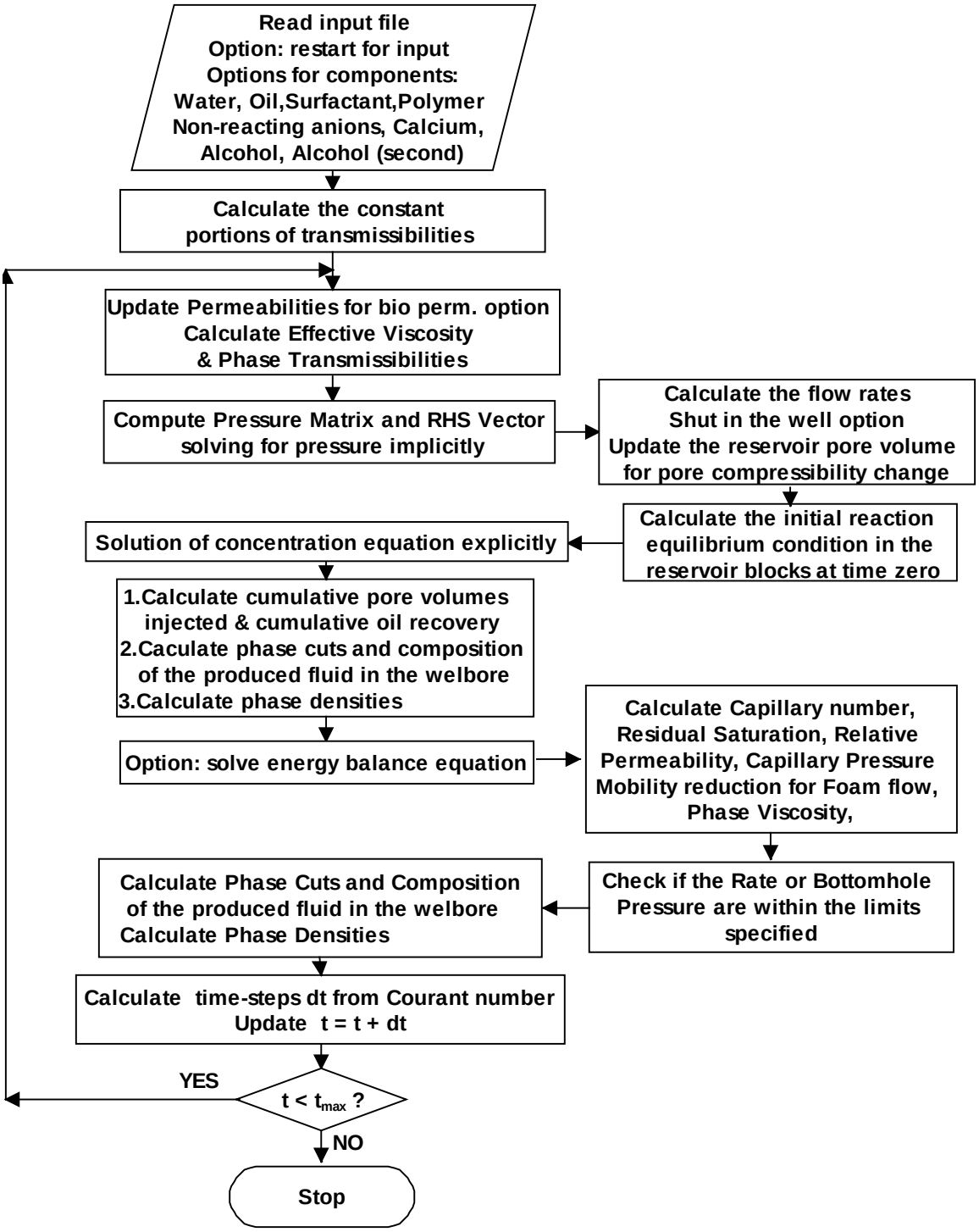

Figure 2.2-1: Flowchart of UTCHEM simulator.

2.3 GPAS Simulator

GPAS is a fully implicit, compositional, equation-of-state, and parallel simulator that contains many modules such as compositional, chemical flooding, thermal, naturally fractured reservoir, geomechanics, asphaltene precipitation, and wellbore modules.

GPAS is a parallel software package written in mixed C and FORTRAN languages, in which the memory management for both parallel and serial structures was written in the C language, and the operation parts were written in the FORTRAN language, hence GPAS can be used for either a single PC (personal computer), massively parallel computers, or a cluster of PCs. The current version of GPAS simulator used in this thesis is Version 3.6; its source code includes 163 files, of which 123 are FORTRAN files, seven are C files, and 33 are head files. The current compiler option is the combination of the Intel FORTRAN compiler with the OpenMPI compiler. GPAS can be compiled, linked and run under both Windows and Linux/Unix operating systems. Its flowchart is presented in Figure 2.3-1.

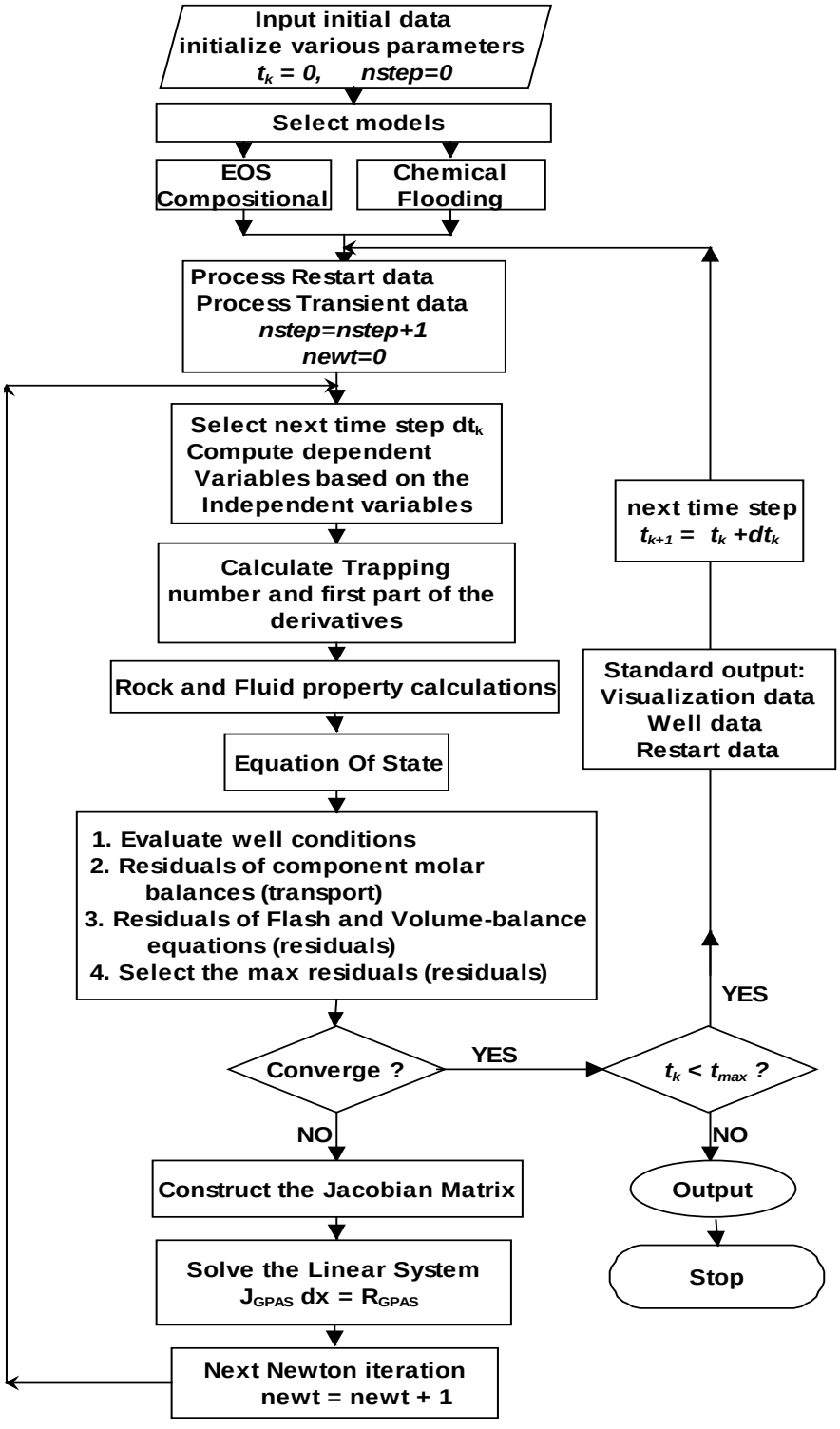

Figure 2.3-1: Flowchart of GPAS.

2.4 HYPRE Software

HYPRE was developed at the Center for Applied Scientific Computing (CASC) at the Lawrence Livermore National Laboratory (LLNL) in the USA, and is an open source software package of high performance preconditioners and solvers for the solution of the sparse linear systems of equations on both a PC (personal computer), massively parallel computers, or a PC cluster. HYPRE can be downloaded from the following website: https://computation.llnl.gov/casc/hypre/software.html. The 2.4.0b version of HYPRY was used in this study. The application of HYPRE code needs to be compiled with a link option to HYPRE library directory.

2.5 SAMG Software

SAMG, developed at Fraunhofer Institute for Algorithms and Scientific Computing SCAI in Germany, is a commercial software package for solving linear systems with an algebraic multigrid approach. SAMG package mainly includes a library file "libamg_xxx-xxxxx_rx.so" and a head file "samg.h." For an application of SAMG, the application code of SAMG just needs to link with SAMG library for the compilation to occur.

Chapter 3: UTCOMP and Linear Solvers

3.1 Mathematical Model Equations of UTCOMP

3.1.1 Governing Equations

The governing fluid flow equations of UTCOMP include four kinds of equations: component molar balance equation, phase-equilibrium relationship equation, volume constraint equation, and phase composition constraint equations (for details see UTCOM-Tech 2003):

$$
\begin{cases}
\textbf{Component Molar Balance} \\
\dfrac{\partial N_i}{\partial t} - V_b \nabla \cdot \left[\sum_{j=1}^{n_p} \xi_j \lambda_j x_{ij} \left(\nabla P_j - \gamma_j \nabla \mathrm{D} \right) + \phi \xi_j S_j \bar{\bar{K}}_{ij} \nabla x_{ij} \right] - q_i = 0 \\
\textbf{Phase Equilibrium Relationship} \\
f_i \left(f_i^1, f_i^2, \ldots, f_i^j \right) = 0; \quad i = 1, 2, \ldots, n_c \quad j = 1, 2, \ldots, n_p - 1) \\
\textbf{Phase Composition Constraint} \\
\sum_{i=1}^{n_c} x_{ij} = 1 \\
\textbf{Volume Constraint} \\
V_b \sum_{i=1}^{n_c} N_i \sum_{j=1}^{n_p} L_j \bar{v}_i - V_p = 0 .
\end{cases}
\qquad (3.1.1\text{-}1)
$$

3.1.2 Pressure Equations

UTCOMP employs an IMPES-like solution procedure, that is, a pressure equation will be implicitly solved, and the component concentration is then solved explicitly. Summing the component molar equation for all components under the assumption that the pore volume should be filled completely by the total fluid volume: $V_t\left(P,\vec{N}\right)=V_P\left(P\right)$, the pressure equation can be obtained as follows:

$$\left(V_p^0 c_f-\frac{\partial V_t}{\partial P}\right)\frac{\partial P}{\partial t}-V_b\sum_{i=1}^{n_c+1}\bar{V}_{ti}\nabla\sum_{j=1}^{n_p}\vec{k}\lambda_{\mathrm{rj}}\xi_{\mathrm{j}}\mathrm{x}_{\mathrm{ij}}\nabla P=$$
$$V_b\sum_{i=1}^{n_c+1}\bar{V}_{ti}\nabla\sum_{j=1}^{n_p}\vec{k}\lambda_{\mathrm{rj}}\xi_{\mathrm{j}}\xi_{\mathrm{sj}}S\left(\nabla P_{c2\,j}-\gamma_j\nabla D\right)+V_b\sum_{i=1}^{n_c+1}\bar{V}_{ti}\nabla\sum_{j=1}^{n_p}{}_{\mathrm{j}}\,{}_{\mathrm{j}}\vec{k}\nabla\mathrm{x}_{\mathrm{ij}}+\sum_{i=1}^{n_c+1}\bar{V}_{ti}q_i\,. \tag{3.1.2-1}$$

3.2 Linear System Equations of UTCOMP

3.2.1 Discretization of Pressure Equation

The linear system of UTCOMP is derived from the discretization of the pressure equation (3.1.2-1). The derivatives in (3.1.2-1) are approximated by the suitable finite difference schemes such as central differencing in space and backward differencing in time. The rest of the coefficients are evaluated at the previous time level, then a linear equations can be obtained from (3.1.2-1) on all gridblocks for the pressure unknown P, which is a non-symmetric and non-positive-definite matrix. Assuming the subscripts n and $n+1$ indicate the time level, and x,y,z indicate the spatial location, then

$$\left(V_p^0 c_f-\frac{\partial V_t}{\partial P}\right)\frac{\partial P}{\partial t}\approx\frac{1}{\Delta t}\left(V_p^0 c_f-\frac{\partial V_t}{\partial P}\right)_{xyz}^{n}\left(P_{xyz}^{n+1}-P_{xyz}^{n}\right). \tag{3.2.1-1}$$

The convection term of (3.1.2-1) can be approximated as follows:

$$\nabla\vec{k}\lambda_{rj}\xi_{xj}\xi_{xj}\nabla P\xi_{xj}=\frac{\partial}{\partial x}\left(k_x\lambda_{rj\ \ j\ \ ij}\frac{\partial P}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_x\lambda_{rj\ \ j\ \ ij}\frac{\partial P}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_x\lambda_{rj\ \ j\ \ ij}\frac{\partial P}{\partial z}\right)$$

$$=\frac{1}{\Delta x}\left[\left(\left(\frac{k_x}{\Delta x}\right)_{x+1/2}\left(\lambda_{rj}\xi_{xj}\xi_{xj}\right)^n_{x+1/2}\left(P^{n+1}_{x+1}-P^{n+1}_{x,}\right)\right)-\left(\left(\frac{k_x}{\Delta x}\right)_{x-1/2}\left(\lambda_{rj\ \ j\ \ ij}\right)^n_{x-1/2}\left(P^{n+1}_{x}-P^{n+1}_{x-1}\right)\right)\right]$$

$$+\frac{1}{\Delta y}\left[\left(\left(\frac{k_y}{\Delta y}\right)_{y+1/2}\left(\lambda_{rj}\xi_{xj}\xi_{xj}\right)^n_{y+1/2}\left(P^{n+1}_{y+1}-P^{n+1}_{y}\right)\right)-\left(\left(\frac{k_y}{\Delta y}\right)_{y-1/2}\left(\lambda_{rj\ \ j\ \ ij}\right)^n_{y-1/2}\left(P^{n+1}_{y}-P^{n+1}_{y-1}\right)\right)\right]$$

$$+\frac{1}{\Delta z}\left[\left(\left(\frac{k_z}{\Delta z}\right)_{z+1/2}\left(\lambda_{rj}\xi_{xj}\xi_{xj}\right)^n_{z+1/2}\left(P^{n+1}_{z+1}-P^{n+1}_{z}\right)\right)-\left(\left(\frac{k_z}{\Delta z}\right)_{y-1/2}\left(\lambda_{rj\ \ j\ \ ij}\right)^n_{z-1/2}\left(P^{n+1}_{z}-P^{n+1}_{z-1}\right)\right)\right].$$

(3.3.1-4)

Full tensor of the physical dispersion term in (3.1.2-2) can be discretized as

$$\nabla\phi\xi_{sj\ \ j}\vec{k}\nabla x_{ij}$$

$$=\frac{\partial}{\partial x}\left[\phi\xi_{sj\ \ j}\left(K_{xx,ij}\frac{\partial x_{ij}}{\partial x}+K_{xy,ij}\frac{\partial x_{ij}}{\partial y}+K_{xz,ij}\frac{\partial x_{ij}}{\partial z}\right)\right]$$

$$+\frac{\partial}{\partial y}\left[\phi\xi_{sj\ \ j}\left(K_{yx,ij}\frac{\partial x_{ij}}{\partial x}+K_{yy,ij}\frac{\partial x_{ij}}{\partial y}+K_{yz,ij}\frac{\partial x_{ij}}{\partial z}\right)\right]$$

$$+\frac{\partial}{\partial z}\left[\phi\xi_{sj\ \ j}\left(K_{zx,ij}\frac{\partial x_{ij}}{\partial x}+K_{zy,ij}\frac{\partial x_{ij}}{\partial y}+K_{zz,ij}\frac{\partial x_{ij}}{\partial z}\right)\right]$$

$$=\frac{1}{\Delta x_x}\left[\begin{array}{l}2\left(\phi\xi_{sj\ \ j}K\xi_{sj}K\right)^n_{x+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+1}-\left(x_{ij}\right)^n_{x}}{\Delta x_x+\Delta x_{x+1}}+\left(\phi_{\ \ j\ \ j\ \ \ xy,ij}\right)^n_{x+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+\frac{1}{2},y+1}-\left(x_{ij}\right)^n_{x+\frac{1}{2},y-1}}{\Delta y_y+\frac{1}{2}\left(\Delta y_{y+1}+\Delta y_{y-1}\right)}\\ +\left(\phi\xi_{sj\ \ j}K_{xz,ij}\right)^n_{x+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+\frac{1}{2},z+1}-\left(x_{ij}\right)^n_{x+\frac{1}{2},z-1}}{\Delta z_z+\frac{1}{2}\left(\Delta z_{z+1}+\Delta z_{z-1}\right)}\end{array}\right]$$

$$-\frac{1}{\Delta x_x}\left[\begin{array}{l}2\left(\phi\xi_{sj\ \ j}K\xi_{sj}K\right)^n_{x-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x}-\left(x_{ij}\right)^n_{x-1}}{\Delta x_x+\Delta x_{x-1}}+\left(\phi_{\ \ j\ \ j\ \ \ xy,ij}\right)^n_{x-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x-\frac{1}{2},y+1}-\left(x_{ij}\right)^n_{x-\frac{1}{2},y-1}}{\Delta y_y+\frac{1}{2}\left(\Delta y_{y+1}+\Delta y_{y-1}\right)}\\ +\left(\phi\xi_{sj\ \ j}K_{xz,ij}\right)^n_{x-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x-\frac{1}{2},z+1}-\left(x_{ij}\right)^n_{x-\frac{1}{2},z-1}}{\Delta z_z+\frac{1}{2}\left(\Delta z_{z+1}+\Delta z_{z-1}\right)}\end{array}\right]$$

$$
\begin{aligned}
&+\frac{1}{\Delta y_y}\left[\begin{array}{l}
2\left(\phi\xi_j K_{yy,ij}\right)^n_{y+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{y+1}-\left(x_{ij}\right)^n_{y}}{\Delta y_y+\Delta y_{y+1}}+\left(\phi_{j\ j\ yx,ij}\right)^n_{y+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+1,y+\frac{1}{2}}-\left(x_{ij}\right)^n_{x-1,y+\frac{1}{2}}}{\Delta x_x+\frac{1}{2}\left(\Delta x_{x+1}+\Delta x_{x-1}\right)}\\
+\left(\phi\xi_j K_{yz,ij}\right)^n_{y-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{y+\frac{1}{2},z+1}-\left(x_{ij}\right)^n_{y+\frac{1}{2},z-1}}{\Delta z_z+\frac{1}{2}\left(\Delta z_{z+1}+\Delta z_{z-1}\right)}
\end{array}\right]\\
&-\frac{1}{\Delta y_y}\left[\begin{array}{l}
2\left(\phi\xi_j K_{yy,ij}\right)^n_{y-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{y}-\left(x_{ij}\right)^n_{y-1}}{\Delta y_y+\Delta y_{y-1}}+\left(\phi\xi_j K_{yx,ij}\right)^n_{y-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+1,y-\frac{1}{2}}-\left(x_{ij}\right)^n_{x-1,y-\frac{1}{2}}}{\Delta x_x+\frac{1}{2}\left(\Delta x_{x+1}+\Delta x_{x-1}\right)}\\
+\left(\phi\xi_j K_{yz,ij}\right)^n_{y-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{y-\frac{1}{2},z+1}-\left(x_{ij}\right)^n_{y-\frac{1}{2},z-1}}{\Delta z_z+\frac{1}{2}\left(\Delta z_{z+1}+\Delta z_{z-1}\right)}
\end{array}\right]\\
&+\frac{1}{\Delta z_z}\left[\begin{array}{l}
2\left(\phi\xi_j K_{zz,ij}\right)^n_{z+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{z+1}-\left(x_{ij}\right)^n_{z}}{\Delta z_z+\Delta z_{z+1}}+\left(\phi_{j\ j\ zx,ij}\right)^n_{z+\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+1,z+\frac{1}{2}}-\left(x_{ij}\right)^n_{x-1,z+\frac{1}{2}}}{\Delta x_x+\frac{1}{2}\left(\Delta x_{x+1}+\Delta x_{x-1}\right)}\\
+\left(\phi\xi_j K_{zy,ij}\right)^n_{y-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{z+\frac{1}{2},y+1}-\left(x_{ij}\right)^n_{z+\frac{1}{2},y-1}}{\Delta y_y+\frac{1}{2}\left(\Delta y_{y+1}+\Delta y_{y-1}\right)}
\end{array}\right]\\
&-\frac{1}{\Delta z_z}\left[\begin{array}{l}
2\left(\phi\xi_j K_{zz,ij}\right)^n_{z-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{z+1}-\left(x_{ij}\right)^n_{z}}{\Delta z_z+\Delta z_{z-1}}+\left(\phi\xi_j K_{zx,ij}\right)^n_{z-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{x+1,z-\frac{1}{2}}-\left(x_{ij}\right)^n_{x-1,z-\frac{1}{2}}}{\Delta x_x+\frac{1}{2}\left(\Delta x_{x+1}+\Delta x_{x-1}\right)}\\
+\left(\phi\xi_j K_{zy,ij}\right)^n_{y-\frac{1}{2}}\dfrac{\left(x_{ij}\right)^n_{z-\frac{1}{2},y+1}-\left(x_{ij}\right)^n_{z-\frac{1}{2},y-1}}{\Delta y_y+\frac{1}{2}\left(\Delta y_{y+1}+\Delta y_{y-1}\right)}
\end{array}\right],
\end{aligned}
\tag{3.2.1-5}
$$

where one-point upstream weight is used to approximate the molar density $\left(\xi_j\right)_{x\pm\frac{1}{2}}$ and phase composition $\left(x_{ij}\right)_{x\pm\frac{1}{2}}$ in the above equation.

The pressure equation is discretized using central differencing in space and backward differencing in time. The finite difference equation of Equation (3.2.1-2) can be obtained as follows:

$$\left(V_p^0 c_f - \frac{\partial V_t}{\partial P}\right)_{xyz}^n P_{xyz}^{n+1} - \left(A_{x+\frac{1}{2}}\,(P_{x+1yz}^{n+1} - P_{xyz}^{n+1}) - A_{x-\frac{1}{2}}\,(P_{xyz}^{n+1} - P_{x-1yz}^{n+1})\right)$$
$$-\left(A_{y+\frac{1}{2}}\,(P_{xy+1z}^{n+1} - P_{xyz}^{n+1}) - A_{y-\frac{1}{2}}\,(P_{xyz}^{n+1} - P_{xy-1z}^{n+1})\right) - \left(A_{z+\frac{1}{2}}\,(P_{xyz+1}^{n+1} - P_{xyz}^{n+1}) - A_{z-\frac{1}{2}}\,(P_{xyz}^{n+1} - P_{xyz-1}^{n+1})\right)$$
$$= \left(V_t - V_p\right)_{xyz}^n + \left(V_p^0 c_f - \frac{\partial V_t}{\partial P}\right)_{xyz}^n P_{xyz}^n - \Delta t \sum_{i=1}^{n_c+1} \left(\bar{V}_{ti}\right)_{xyz}^n q_i$$
$$+\Delta t \left(\sum_{i=1}^{n_c+1} \left(\bar{V}_{ti}\right)_{xyz}^n \left(\sum_{j=1}^{n_p} \Delta\left(\mathrm{x}_{\mathrm{ij}} T_j\right)^n \Delta P_{coj}^n - \sum_{j=1}^{n_p} \Delta\left(\mathrm{x}_{\mathrm{ij}} T_j \gamma_j\right)^n \Delta D - \left(V_b\right)_{xyz} \sum_{j=1}^{n_p} \Delta\left(J_{xij}^n + J_{yij}^n + J_{zij}^n\right) \right) \right). \quad (3.2.1\text{-}6)$$

Rearrange of (3.2.1-6), we can

have

$$\left[\left(V_p^0 c_f - \frac{\partial V_t}{\partial P}\right)_{xyz}^n + \left(A_{x+\frac{1}{2}} - A_{x-\frac{1}{2}}\right) + \left(A_{y+\frac{1}{2}} - A_{y-\frac{1}{2}}\right) + \left(A_{z+\frac{1}{2}} - A_{z-\frac{1}{2}}\right)\right] P_{xyz}^{n+1}$$
$$-A_{x+\frac{1}{2}} P_{x+1yz}^{n+1} - A_{x-\frac{1}{2}} P_{x-1yz}^{n+1} - A_{y+\frac{1}{2}} P_{xy+1z}^{n+1} - A_{y-\frac{1}{2}} P_{xy-1z}^{n+1} - A_{z+\frac{1}{2}} P_{xyz+1}^{n+1} - A_{z-\frac{1}{2}} P_{xyz-1}^{n+1}$$
$$= \left(V_t - V_p\right)_{xyz}^n + \left(V_p^0 c_f - \frac{\partial V_t}{\partial P}\right)_{xyz}^n P_{xyz}^n - \Delta t \sum_{i=1}^{n_c+1} \left(\bar{V}_{ti}\right)_{xyz}^n q_i$$
$$+\Delta t \left(\sum_{i=1}^{n_c+1} \left(\bar{V}_{ti}\right)_{xyz}^n \left(\sum_{j=1}^{n_p} \Delta\left(\mathrm{x}_{\mathrm{ij}} T_j\right)^n \Delta P_{coj}^n - \sum_{j=1}^{n_p} \Delta\left(\mathrm{x}_{\mathrm{ij}} T_j \gamma_j\right)^n \Delta D - \left(V_b\right)_{xyz} \sum_{j=1}^{n_p} \Delta\left(J_{xij}^n + J_{yij}^n + J_{zij}^n\right) \right) \right), \quad (3.2.1\text{-}7)$$

where $A_{m\pm\frac{1}{2}}$ in Equation (3.2.1-7) expresses in Equation (3.2.1-8):

$$A_{m\pm\frac{1}{2}} = \Delta t \sum_{i=1}^{n_c+1} \left(\bar{V}_{ti}\right)_m^n \sum_{j=1}^{n_p} \left(\mathrm{x}_{\mathrm{ij}} T_j\right)_{m\pm\frac{1}{2}}^n, m = x, y, z. \quad (3.2.1\text{-}8)$$

3.2.2 The Linear Systems for Pressure Equation

The linear system of equations representing the discretized form of the pressure equation in UTCOMP can be obtained from (3.3.1-7) together with calculation of fluid mass and molar density and volume, viscosity, capillary pressure, relative permeability and well models as follows:

$$Au = RHS, \tag{3.2.2-1}$$

where matrix A has a heptagonal sparse pattern shown in Figure 3.2.2-1 as an example of 4x4x3 girdblocks, which has the following properties:

Matrix A is not symmetric. The nonzero pattern of A is symmetric, so we cannot judge whether A is symmetric or not. In order to show that A is not symmetric for this case, we select some nonzero elements of value "77.2397284081030" for this case, then each nonzero element of A subtracts the value "77.2397284081030" to get a new nonzero pattern matrix shown in Figure 3.2.2-2, in which the red-circle-shape elements in Figure 3.2.2-2 are the new zero elements compared to Figure 3.2.2-1. We can see the new nonzero pattern in Figure 3.2.2-2 is not symmetric, so we can infer that A is not a symmetric matrix (the proof is trivial).

The structure of the nonzero pattern of matrix A from UTCOMP shows the geometric information of the region. Since the gridblock numbering in UTCOMP start from Y-direction fist, Z-direction second, and X-direction last (YZX order), the structure of nonzero pattern in Figure 3.2.2-1 represents the result of the approaches of YZX order numbering from UTCOMP.

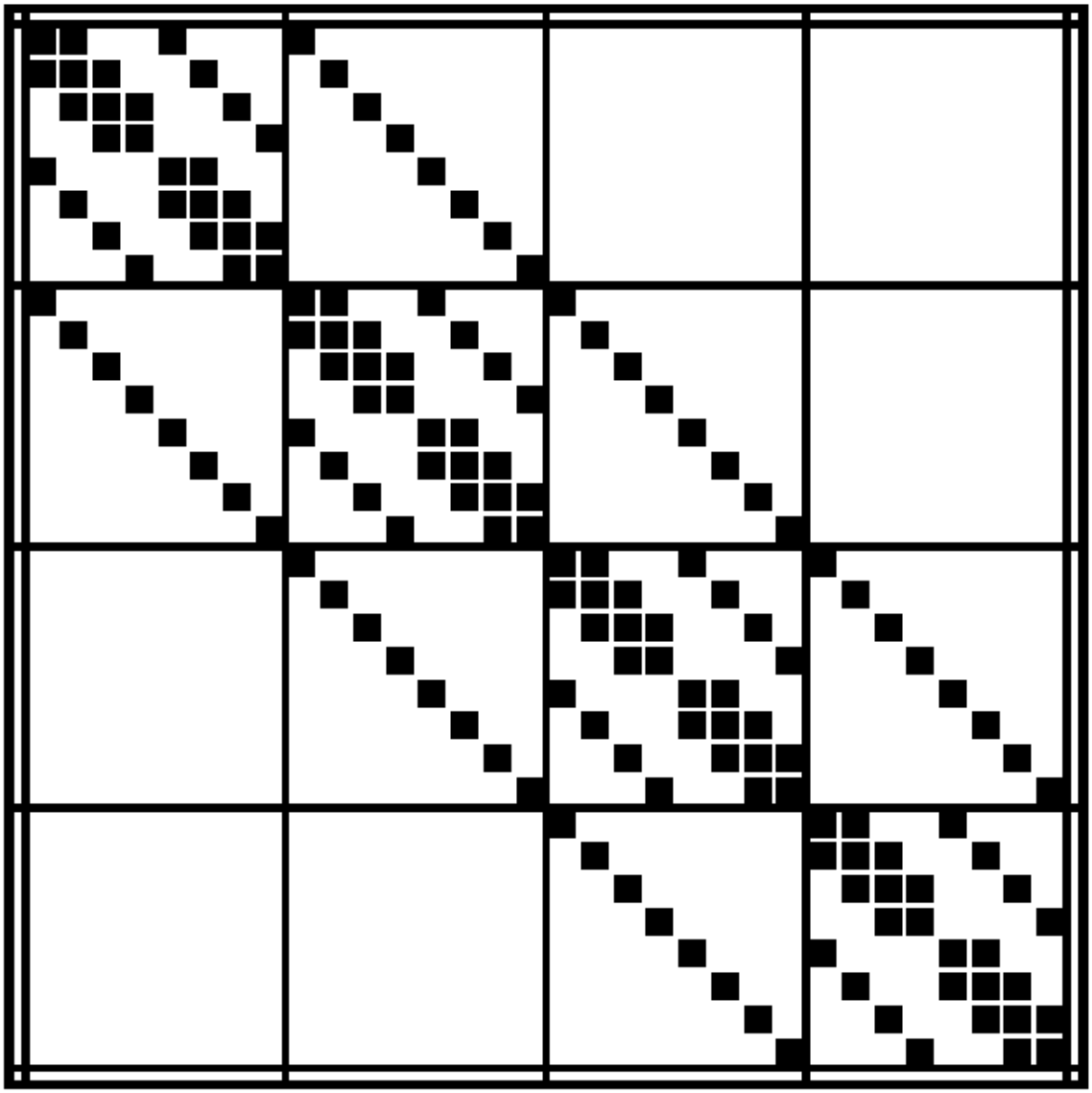

Figure 3.2.2-1: Heptagonal sparse pattern for 4x4x3 gridblocks from UTCOMP.

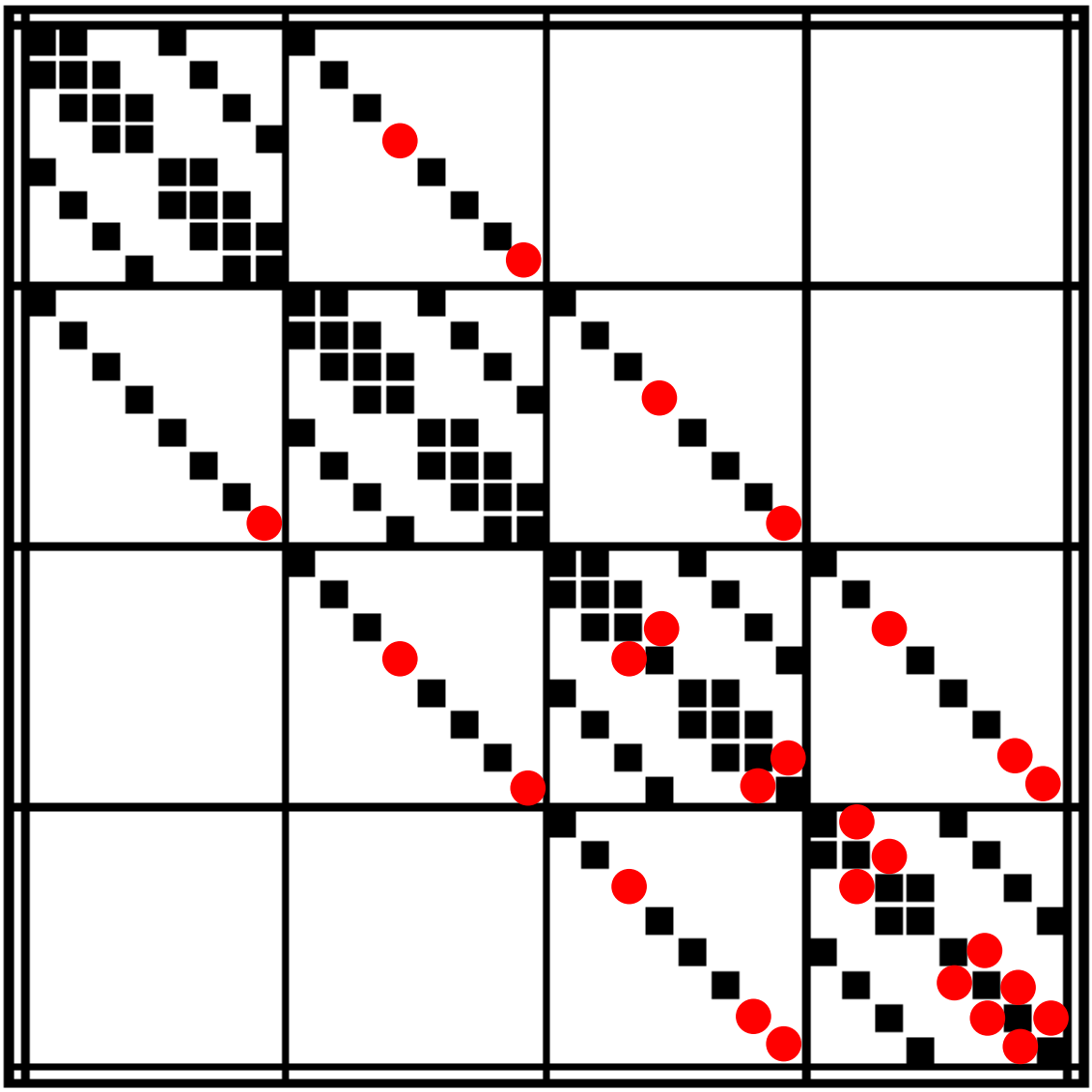

Figure 3.2.2-2: New heptagonal nonsymmetrical sparse pattern from Figure3.2.2-1.

3.3 Linear Solvers in UTCOMP

The existing solvers in UTCOMP include three iterative methods and four preconditioner options. The iterative methods are BiCG (Biconjugate Gradient) iteration method, BiCGS/CGS (Conjugate Gradient Squared) iteration method, and ORTHOMIN (Truncated/restarted ORTHORMIN) iteration method. The preconditioner options are left preconditioners consisting of Richardson method, Jacobi preconditioner, ILU decomposition preconditioner, and modified ILU decomposition preconditioner.

3.3.1 Iterative Methods in UTCOMP

BiCG: The conjugate gradient (CG) method cannot solve nonsymmetric systems because the residual vectors cannot be made orthogonal with short recurrences, which was shown by Voevodin (1983) and Faber et al. (1984). One way to improve CG is the generalized minimal residual method which retains orthogonality of the residuals using long recurrences, at the cost of a larger storage demand. The biconjugate gradient method (BiCG) is another approach, replacing the orthogonal sequence of residuals by two mutually orthogonal sequences, at the cost of no longer providing a minimization.

The update relations for residuals in the CG are augmented in the BiCG by relations that are similar but based on A^T instead of A. Thus, we update two sequences of residuals in Equation (3.3.1-1)

$$r^{(i)} = r^{(i-1)} - \alpha_i A p^{(i)}; \quad \overline{r}^{(i)} = \overline{r}^{(i-1)} - \alpha_i A^T \overline{p}^{(i)}, \tag{3.3.1-1}$$

and two sequences of search directions in Equation (3.3.1-2)

$$p^{(i)} = r^{(i-1)} + \beta_{i-1} p^{(i-1)}; \overline{p}^{(i)} = \overline{r}^{(i-1)} + \beta_{i-1} \overline{p}^{(i-1)}, \tag{3.3.1-2}$$

where

$$\alpha_i = \frac{\overline{r}^{(i-1)^T} \overline{r}^{(i-1)}}{\overline{p}^{(i)^T} A p^{(i)}}, \quad \beta_i = \frac{\overline{r}^{(i)^T} \overline{r}^{(i)}}{\overline{r}^{(i-1)^T} \overline{r}^{(i-1)}}, \tag{3.3.1-3}$$

ensure the orthogonality relations Equation (3.3.1-4)

$$\bar{r}^{(i)^T} \bar{r}^{(j)} = \bar{p}^{(i)^T} A p^{(j)} = 0 \quad \text{for} \quad i \neq j . \tag{3.3.1-4}$$

Few theoretical results are known about the convergence of BiCG (Sonneveld 1989). For symmetric positive definite systems, the method delivers the same results as CG, but at twice the cost per iteration. For non-symmetric matrices, it has been shown that in phases of the process where there is significant reduction of the norm of the residual, the method is more or less comparable to the full generalized minimal residual method in terms of number of iterations (Freund et al. 1991). In practice, this is often confirmed, but it is also observed that the convergence behavior may be quite irregular, and the method may even break down. Due to the possible cases, the breakdown situation that

$$z^{(i-1)^T} r^{(i-1)} \approx 0 , \tag{3.3.1-5}$$

can be circumvented by so-called look-ahead strategies (Parlett et al. 1985). The other breakdown situation,

$$p^{(i)^T} q^{(i)} \approx 0 , \tag{3.3.1-6}$$

occurs when the LU decomposition fails and can be repaired by using another decomposition. The flowchart of BiCG used in UTCOMP is shown in Figure 3.3.1-1.

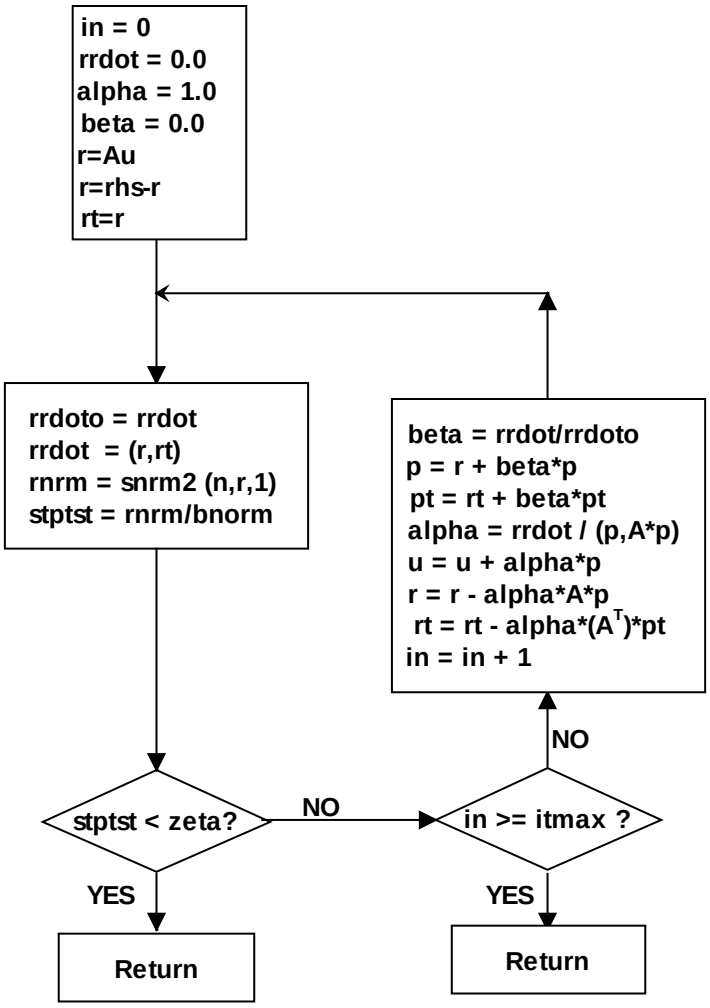

Figure 3.3.1-1: Flowchart of BiCG used in UTCOMP.

BiCGS/CGS: In BiCG, the residual vector, $r^{(i)}$, can be regarded as the product of $r^{(0)}$ and an i^{th} degree polynomial in A, i.e., $r^{(i)} = p_i(A)r^{(0)}$. The same polynomial satisfies $\overline{r}^{(i)} = p_i(A^T)\overline{r}^{(0)}$ so that

$$\rho^{(i)} = (\overline{r}^{(i)}, r^{(i)}) = (p_i(A)\overline{r}^{(0)}, p_i(A)r^{(0)}) = (\overline{r}^{(0)}, p_i^2(A)r^{(0)}) \qquad (3.3.1\text{-}7)$$

Equation (3.3.1-8) suggests that if $p_i(A)$ reduces $r^{(0)}$ to a smaller vector $r^{(i)}$, then it might be advantageous to apply this "contraction" operator twice, and compute $p_i^2(A)r^{(0)}$. The iteration coefficients can still be recovered from these vectors (as shown in Equation

(3.3.1-7)), and it turns out to be easy to find the corresponding approximations for x. This approach is the biconjugate/conjugate gradient squared (BiCGS/CGS) method (Sonneveld 1989).

Often one observes a speed of convergence for BiCGS that is about twice as fast as that for the BiCG, which is in agreement with the observation that the same "contraction" operator is applied twice. However, there is no reason that the contraction operator, even if it really reduces the initial residual $r^{(0)}$, should also reduce the once reduced vector $r^{(k)} = p_k(A)r^{(0)}$. This is evidenced by the often highly irregular convergence behavior of BiCGS. One should be aware of the fact that local corrections to the current solution may be so large that cancellation effects occur.

BiCGS requires about the same number of operations per iteration as BiCG, but does not involve computations with A^T. Hence, in circumstances where computation with A^T is impractical, BiCGS may be attractive. The flowchart of BiCGS in UTCOMP is shown in Figure 3.3.1-2.

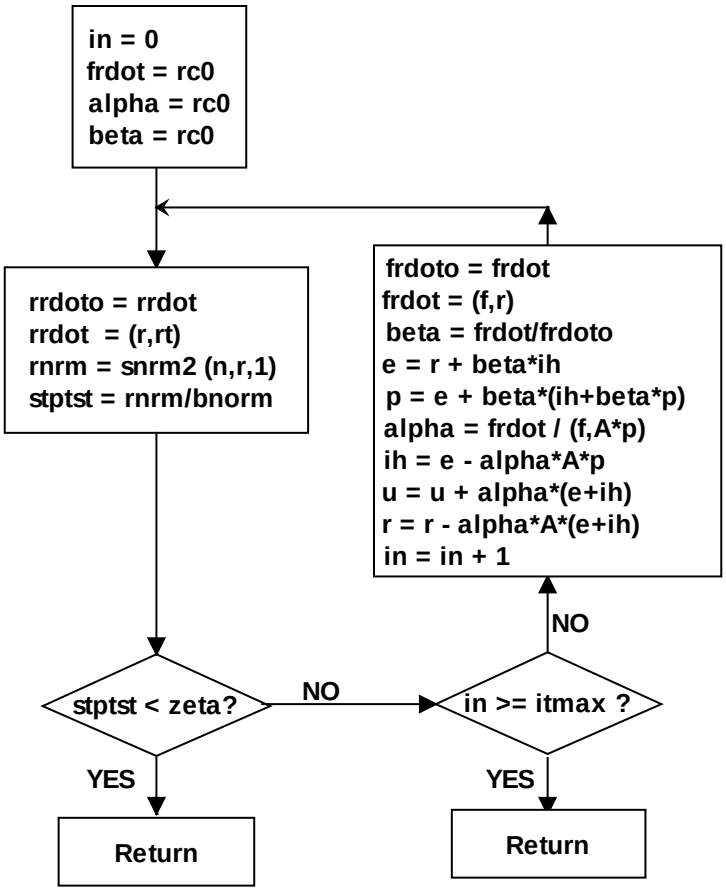

Figure 3.3.1-2: Flowchart of BiCGS used in UTCOMP.

Truncated/Restarted ORTHORMIN: ORTHORMIN was proposed for the solution of spare band-structured matrix by Vinsome (1976). Based on the conjugate-gradient method of solving symmetric matrices linear systems with a minimization process, ORTHORMIN was developed for non-symmetric spars matrices by using both orthorgonalisations and minimizations to achieve a high rate of convergence. The ORTHORMIN coded in UTCOMP is presented in Figure 3.3.1-3.

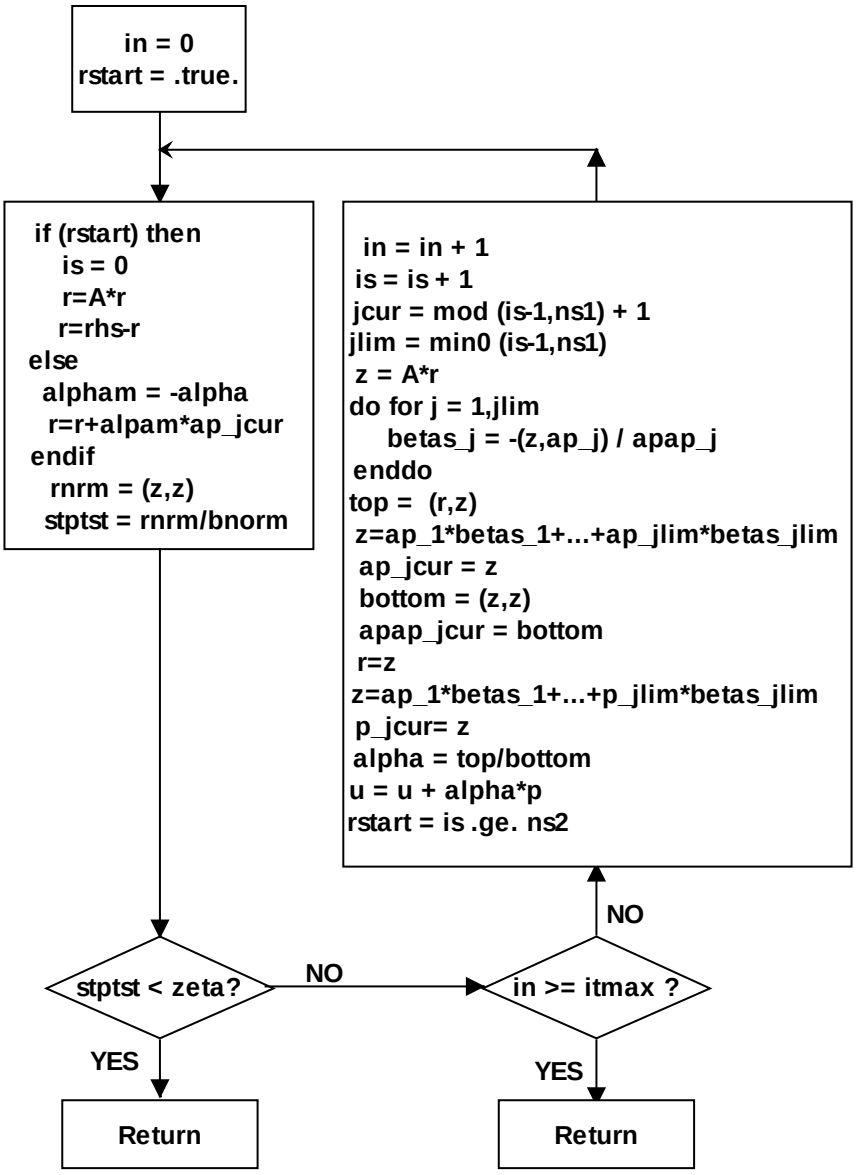

Figure 3.3.1-3: Flowchart of ORTHOMIN used in UTCOMP.

3.3.2 Preconditioners in UTCOMP

The choice of a preconditioner involves the selection of a matrix M, called the splitting matrix, such that the preconditioned system

$$M^{-1}Au = M^{-1}RHS \qquad (3.3.2\text{-}1)$$

is better conditioned than the original system ($Au = RHS$). The left preconditioners in Equation (3.3.2-1) were applied to the linear solver in UTCOMP.

There are three preconditioners in UTCOMP such as Jacobi, incomplete LU (ILU) decomposition, and modified incomplete LU (MILU) decomposition preconditioners.

We assume that matrix A in Equation (3.3-1) has the following form

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{21} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \qquad (3.3.2\text{-}2)$$

Jacobi preconditioner: The Jacobi preconditioner is constructed from matrix A in Equation (3.3.2-2) as follows:

$$M = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix} \square\ diag(a_{11}, a_{22}, \cdots, a_{nn}). \qquad (3.3.2\text{-}3)$$

Incomplete LU decomposition preconditioner: This preconditioner uses an incomplete LU(k) decomposition of the matrix as a preconditioner. The parameter k denotes the level of fill-in which is to be allowed in the factorization. The form of M is

$$M = \begin{cases} (\Delta - C_L)\Delta^{-1}(\Delta - C_U) & if\ case\ I \\ (\Delta - S)\Delta^{-1}(\Delta - T) & if\ case\ II \end{cases}. \qquad (3.3.2\text{-}4)$$

Case I occurs when matrix A is a circulant Toeplitz matrix and $k = 0$. Case II occurs if the condition of Case I fails. Here, Δ is a diagonal matrix containing the factorization pivots, S is a strictly lower triangular matrix, and T is a strictly upper triangular matrix.

It can be seen that if Case I is true, then a considerable saving in storage is possible since only a vector of pivots of length N has to be stored. M can then be implicitly represented from just Δ and the given matrix elements, which are already stored. If Case II is true, then it is necessary to store T as well S, if A is non-symmetrical.

A fill-in level of $k = 0$ means that no fill-in beyond the original matrix pattern of nonzero is to be allowed in the factorization. For $k = 1$, fill-in resulting from the original nonzero pattern is allowed but no fill-in resulting from this newly created fill-in is allowed. In general, fill-in at level k results from fill-in from levels $0,1,2,...,k-1$. As k grows, the number of iterations should decrease but at the expense of increased storage and time per iteration.

Modified incomplete LU decomposition preconditioner: This factorization is similar to the ILU(k) preconditioner except that the diagonal pivots of the factorization are adjusted so that $M^{-1}A$ has zero row sums. For many matrices, this requirement produces a better condition number for $M^{-1}A$ than for the ILU(k) preconditioner. Also, this requirement forces $M^{-1}A$ to have at least one eigenvalue equal to one. As in the previous preconditioner, a variable level of fill-in is allowed.

3.4 Implementation of BoomerAMG and SAMG in UTCOMP

3.4.1 Options of BoomerAMG and SAMG in Flowchart

The implementations of SAMG and BoomerAMG were added into UTCOMP at the subroutine where existing linear solvers were called. The location of solver options for BoomerAMG and SAMG in UTCOMP is shown in Figure 3.4.1-1.

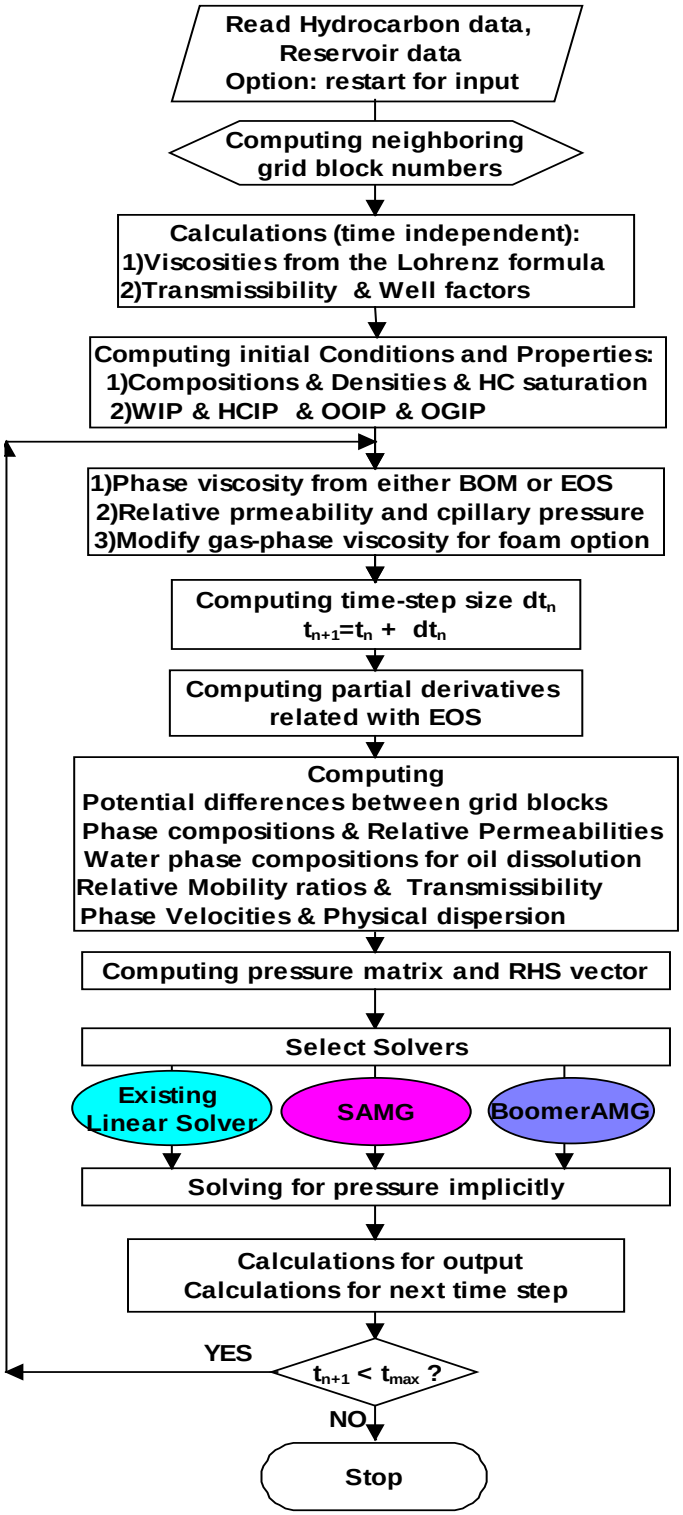

Figure 3.4.1-1: Flowchart of UTCOMP with new solvers.

3.4.2 Implementation of Both Solvers

The first step of the implementation is adding indexes that represent the options for both new linear solvers. In “solve.f” file, IPRESS equals 6 or 7 and IACCEL equals 4 or 5 for SAMG and BoomerAMG respectively (Figure 3.4.2-1).

```
IF (IPRESS .EQ. 3) IACCEL = 1
IF (IPRESS .EQ. 4) IACCEL = 2
IF (IPRESS .EQ. 5) IACCEL = 3
IF (IPRESS .EQ. 6) IACCEL = 4
IF (IPRESS .EQ. 7) IACCEL = 5
```

Figure 3.4.2-1: Option indexes for new solvers used in UTCOMP.

The second step is that calling new solvers will be carried out when an index of new solvers is selected. In “iter.f” file, UTCCOMP calls interface subroutines: SAMG0 and hypre0 of SAMG and BoomerAMG when IACCEL = 4 or 5, the same parameters which were passed to the existing linear solver will pass to the interface subroutines of two new solvers.

```
IF (IACCEL .EQ. 4)then
   CALL SAMG0(N,PNEW,BNEW,WKSP,NW,IPARM,RPARM,IER)
   CALL POST (N)
   RETURN
endif
IF (IACCEL .EQ. 5)then
   CALL HYPRE0(N,PNEW,BNEW,WKSP,NW,IPARM,RPARM,IER)
   CALL POST (N)
   RETURN
endif
```

Figure 3.4.2-2: New options in “iter.f” used in UTCOMP.

In the interface subroutines SAMG0 and hypre0, there are three steps for connecting the new solvers. The first is converting matrix data structure into the format of SAMG and BoomerAMG. The second is passing parameters of UTCOMP to the new

solvers, and setting parameters for the new solver. The third is calling the new solvers (Figure 3.4.3-3).

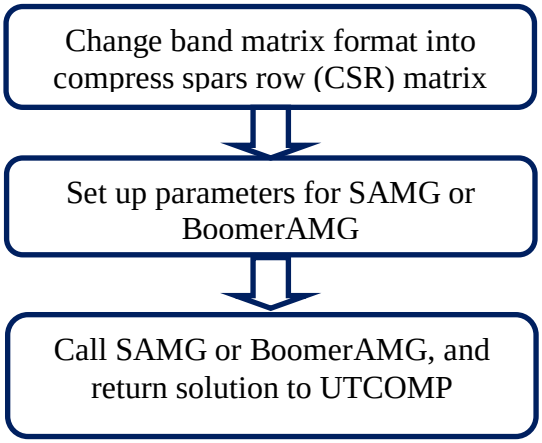

Figure 3.4.3-3: Connecting SAMG and BoomerAMG with UTCOMP.

3.5 Numerical Results and Discussions

In order to test the efficiency of all linear solvers that have been implemented in UTCOMP, three kinds of tests were conducted in this section. The tests were simulation of solvent injections, CO_2 injection, and water injection. Based on our experience, BCG is the fastest iterative method among BCG, BCGS, and OTHORMIN, so we only used BCG to compare the new solvers in this section. Unfortunately, BoomerAMG did not converge completely for the cases using UTCOMP, so there are no numerical results for BoomerAMG in this section.

There are three preconditioners such as Jacobi, ILU(0) and modified ILU(0) together with BCG for the numerical comparison here. BCG32 refers to BCG with Jacobi preconditioner; BCG33 refers to BCG with ILU0 preconditioner; BCG34 refers to BCG with modified ILU0 preconditioner. The simulation period is 100 days. SAMG_10_1 means that SAMG does one “setup” and keeps it for the next nine time-step iterations, which will use it without any cost to “setup” phase.

3.5.1 Solvent Injection Test (Case I)

A hypothetical two-solvent-injection process in a homogeneous reservoir is selected. Reservoir properties and some parameters used in simulation are shown in Table 3.5.1-1.

Table 3.5.1-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Formation temperature	150 °F
Formation compressibility	0.50E-6
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.25
Porosity	0.3
Permeability	100 md
Number of gridblocks (NX*NY*NZ)	140x140x5
Grid block size	$75\times75\times50$ ft^3
Number of hydrocarbon components	6
Component names	C_1,C_3,C_6,C_{10},C_{15},C_{20}
Maximum number of phases	3
Number of wells (Injection/Production)	2 (1/1)
Simulation time	100 days
Injection rate	5000 mscf/day
Injection contents	80% C_1, 20% C_3
Bottomhole production pressure	3100 psi

Figure 3.5.1-1 shows the reservoir region along with injection and production wells (a quarter of a 5-spot pattern). The injection well injected the gas consisting of 80% C_1 and 20% C_3 with a constant injection pressure of 5,000 psi. A producer constraint was assigned to be a constant bottomhole pressure of 3,100 psi.

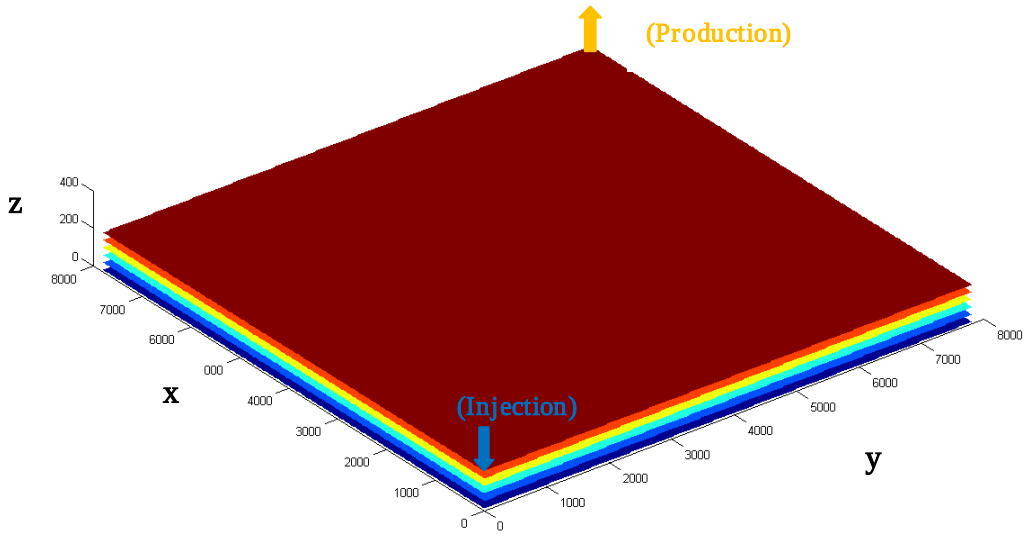

Figure 3.5.1-1: Reservoir model for Case I.

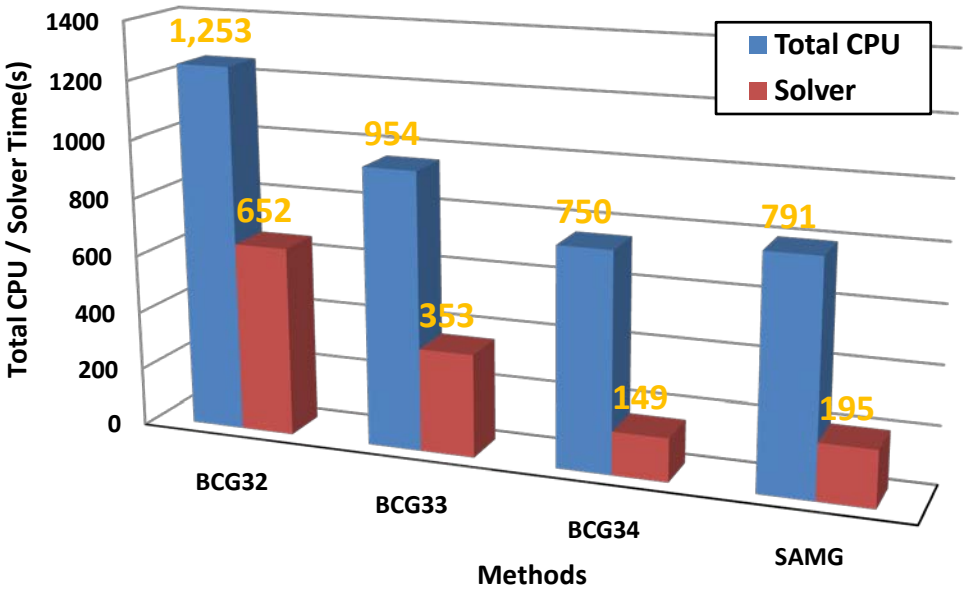

Figure 3.5.1-2: Comparison of running times for Case I.

Figure 3.5.1-2 shows the solver and CPU times spent for running Case I using UTCOMP with four solvers, BCG32, BCG33, BCG34, and SAMG. Case I has a shorter simulation time and a smaller number of gridblocks compared to other cases. The results show BCG34 is the fastest solver among them, and SAMG is second, but its efficiency is much closer to that of BCG32 than that of BCG33 and BCG32.

3.5.2 CO_2 Injection Test (Case II)

A CO_2 injection process is chosen for this case. Reservoir properties and some parameters used in this simulation are shown in Table 3.5.2-1.

Table 3.5.2-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Formation temperature	80 °F
Formation compressibility	3.0E-6
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.25
Porosity	0.3
Permeability	100 md
Number of gridblocks (NX*NY*NZ)	195×195×5
Grid block size	40×40×40 ft^3
Number of hydrocarbon components	3
Component's names	C_1, NC_{16} , CO_2
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	700 days
Initial reservoir pressure	3000 psi
Injection rate	20,000 mscf/day
Injection contents	95% CO_2, 5% C_1
Bottomhole production pressure	3000 psi

The injection well injected the gas consisting of 95% CO_2 and 5% C_1 with a constant injection pressure of 20,000 psi. The producer's constraint is at a constant bottomhole pressure of 3000 psi.

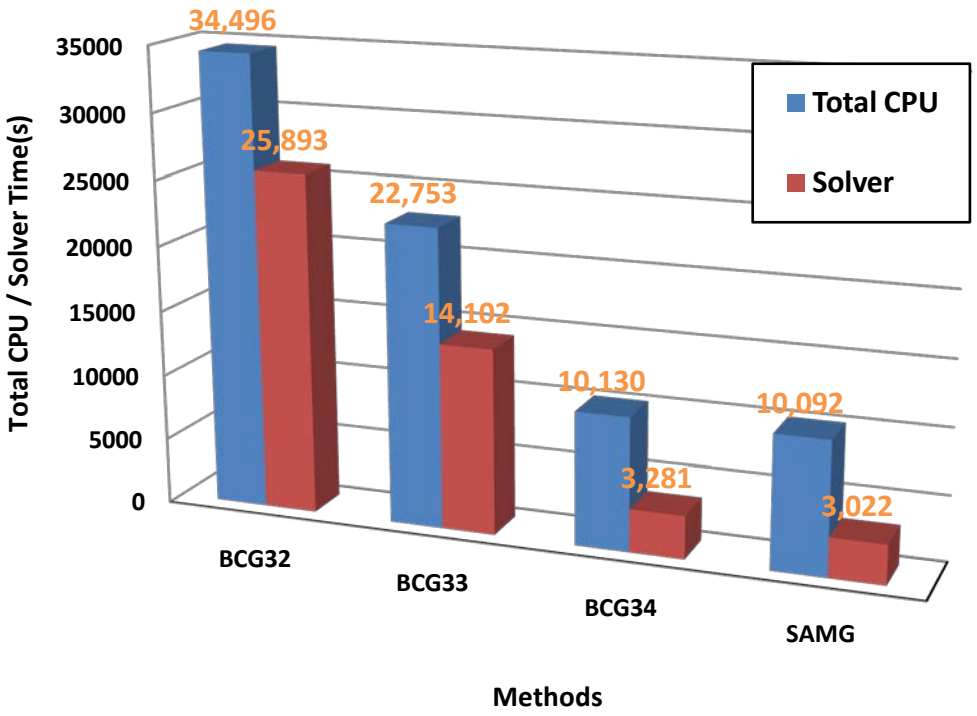

Figure 3.5.2-2: Comparison of running times for Case II.

Figure 3.5.2-2 shows the solver and CPU times for Case II using UTCOMP for four solvers: BCG32, BCG33, BCG34, and SAMG. Case II has a larger reservoir region than that of Case I and less number of components. The solver time of SAMG is less than 92% that of BCG34, less than 21% of that of BCG33, and less than 11% of that of BCG32. Figure 3.5.2-2 shows that SAMG is the fastest solver, and BCG34 is faster than BCG33 and BCG32.

3.5.3 Water Injection Test (Case III)

A water injection test was conducted next. The linear solver's tolerance was set to 10^{-14}.

Reservoir properties and simulation parameters are presented in Table 3.5.3-1.

Table 3.5.3-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Formation temperature	60 °F
Formation compressibility	0.0
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.2
Porosity	0.2
Permeability in the X direction	1000 md
Permeability in the Y direction	1000 md
Permeability in the Z direction	100 md
Number of gridblocks (NX*NY*NZ)	195×195×5
Grid block size	40×40×10 ft^3
Number of hydrocarbon component	1
Component's names	C_{10}
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	600 days
Initial reservoir pressure	200 psi
Injection rate	17500std/day
Injection contents	100% H_2O
Bottomhole production pressure	200 psi

The injection well injected 100% water with a constant injection rate of 3500 STB/day.

The producer's constraint was set at a constant bottomhole pressure of 200 psi.

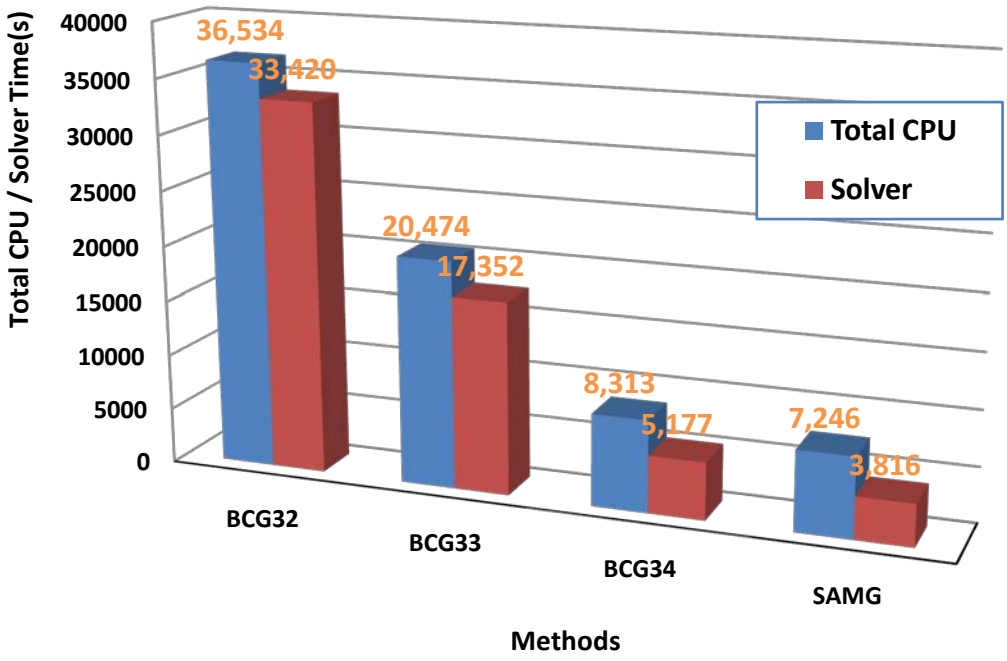

Figure 3.5.3-1: Comparison of running times for Case III.

Figure 3.5.3-2 shows solver and CPU times spent for Case III using UTCOMP for four solvers. Running times in Figure 3.5.3-2 shows that SAMG is the fastest solver, which took less than 74% of running time than BCG34, and less than 21% of the running time of BCG33, less than 11% of running time of BCG32. BCG34 is the second fastest solver, which took less than one-third of BCG33's running time, and one-seventh of BCG32's running time.

3.5.4 Comments and Conclusions

From the previous test results, we may draw the following conclusions:

1) According to both solver and CPU times, SAMG is with a constant injection for Case II and Case III, but is the second fastest solver for Case I. BCG34 is the fastest solver for Case I and the second fastest solver for Case II and Case III. The remaining two solvers BCG33 and BCG34 are at least three times slower than SAMG and BCG34.
2) For the case with shorter simulation time and the smaller number of gridbloks compared to other cases, i.e., Case I, BCG34 is faster than SAMG.
3) Comparisons show the additional advantage of SAMG because SAMG uses original residue $\|R\|_2 = \|b - Ax\|_2$ to achieve the convergence criteria; however, BCG32, BCG33 and BCG34 uses modified residue $\|M^{-1}R\|_2 = \|M^{-1}(b - Ax)\|_2$ to achieve the convergence criteria due to their left preconditioner. Usually $\|M^{-1}R\|_2 < \|R\|_2$ for the diagonal dominant matrix A for the reservoir problems, thus, BCG32, BCG33 and BCG34 had an unfair advantage, that is, the tolerances that were used for BCG32, BCG33, and BCG34 are higher than SAMG's tolerances for the same test problems. However SAMG is still the fastest.
4) For small-scale problems such as problem with the number of gridblocks is less than 1000 and low accuracy requirement problems with the linear solver's tolerance bigger than 10^{-5}, BCG32, BCG33 and BCG34 are better options than SAMG because SAMG will spend a large percentage of solver time on its "setup phase." Therefore, SAMG will have low efficiency.
5) For the large problems with high accuracy requirements, SAMG is a better option than other solvers used in this study.

Chapter 4: UTCHEM and Linear Solvers

4.1 Mathematical Model Equations of UTCHEM

4.1.1 Governing Equations of UTCHEM

There are three governing equations for UTCHEM based on Delshad et al. (1996), consisting of the mass conservation equations, the energy conservation equation, and the nonequilibrium dissolution equation as shown below: (For details see UTCHEM-Volume I 2003)

$$\begin{cases} \textbf{Mass Balance Equations} \\ \dfrac{\partial}{\partial t}\left(\phi \tilde{C}_k \rho_k\right) + \nabla \cdot \left[\sum_{l=1}^{n_p} \rho_k \left(C_{kl}\vec{u}_l - \vec{\tilde{D}}_{kl}\rho_k\right)\right] = \phi \sum_{l=1}^{n_p} S_l r_{kl} + (1-\phi) r_{ks} + Q_k \\ \textbf{Nonequilibrium Dissolution Equations} \\ \dfrac{\partial\left(S_l C_{2l}\phi\right)}{\partial t} = \nabla \cdot \left(C_{2l}\vec{u}_l - \vec{\tilde{D}}_{2l}\right) + M_2\left(C_{2l}^{eq} - C_{2l}\right) \\ \textbf{Energy Balance Equations} \\ \dfrac{\partial}{\partial t}\left[(1-\phi)\rho_s C_{vs} + \phi \sum_{l=1}^{n_p} \rho_l S_l C_{vl}\right] + \nabla \cdot \left[\sum_{l=1}^{n_p} \rho_k C_{pl} u_l T - \lambda_T \nabla T\right] = q_H - Q_L \,. \end{cases} \quad (4.1.1\text{-}1)$$

4.1.2 Aqueous Phase Pressure Equation

UTCHEM is an IMPES-like solution procedure, which needs pressure unknowns to be solved in an implicit approach. The pressure equation can be obtained through summing the mass balance equations over all components together with Darcy's law for the phase flux terms, and simplifying with $\sum_{k=1}^{n_{cv}} C_{kl} = 1$. The pressure equation in terms of reference

pressure has the following form in Equation (4.1.2-1) (For details see UTCHEM-Volume I 2003):

$$\phi C_t \frac{\partial P_R}{\partial t} + \nabla \cdot \sum_{l=1}^{n_p} \vec{u}_l (1 + \Delta P_R \sum_{k=1}^{n_{cv}} C_k^0 C_{kl})_1 = \sum_{k=1}^{n_{cv}} q_k \, . \qquad (4.1.2\text{-}1)$$

4.2 Linear System Equations of UTCHEM

4.2.1 Discretization of Pressure Equation

The pressure equation and species conservation equations are discretized spatially and temporally. The temporal discretization in UTCHEM is implicit in pressure and explicit in concentration (IMPES-like). The implicit solution of the pressure equation is then followed by a back substitution into the explicit mass conservation equation for each component. The temporal accuracy for the conservation equation is increased by using a time-correction technique that is a second order in time (Liu, 1993; Liu et al., 1994).

Either one-point upstream, two-point upstream, or a third-order spatial discretization of the advective terms is used. It is well-known that lower-order upwind schemes cause smearing of the saturation and concentration profiles by increasing numerical dispersion. UTCHEM employs a scheme that is approximately a third-order in space to minimize numerical dispersion and grid-orientation effects.

From the discretization of pressure equation (4.1.3-5), we have

$$\left(F_a\right)_m^{n+1} + \delta_t t^n \left(F_t + F_q\right)_m^{n+1} / 2 = \left(F_a\right)_m^n - \delta_t t^n \left(F_t + F_q\right)_m^n / 2 \quad , \qquad (4.2.1\text{-}1)$$

where the total accumulation F_a in Equation (4.2.1-1) is

$$\left(F_a\right)_m = \left(\phi_R \Delta V\right)_m \left\{ C_f + \sum_{k=1}^{n_{cv}} C_k^0 \left[\left(1 - \sum_{k=1}^{n_{cv}} \hat{\hat{C}}_k \right) \sum_{l=1}^{n_p} S_l C_{kl} + C_k \right] \right\}_m \left(P_R\right)_m , \qquad (4.2.1\text{-}2)$$

and the total transport F_t in Equation (4.2.1-1) is

$$\begin{aligned}\left(F_t\right)_m = &-\delta_x \left\{ \begin{array}{l} \sum_{l=1}^{n_p} \overline{T}_{xl} \left[1+\left(P_R - P_{R0}\right) \sum_{k=1}^{n_{cv}} C_k^0 C_{kl} \right]_m \\ \left[\delta_x \left(P_R + P_{clR}\right)_{m+1} - \left(\overline{\gamma}_{xl}\right)_m \delta_x \left(D\right)_{m+1} \right] \end{array} \right\} \\ &-\delta_y \left\{ \begin{array}{l} \sum_{l=1}^{n_p} \overline{T}_{yl} \left[1+\left(P_R - P_{R0}\right) \sum_{k=1}^{n_{cv}} C_k^0 C_{kl} \right]_m \\ \left[\delta_y \left(P_R + P_{clR}\right)_{m+1} - \left(\overline{\gamma}_{yl}\right)_m \delta_x \left(D\right)_{m+1} \right] \end{array} \right\}, \end{aligned} \tag{4.2.1-3}$$

also the total source and sink F_q in Equation (4.2.1-1) is

$$\left(F_q\right)_m = \sum_{l=1}^{n_p} \left[Q_l + PI_l \left(P_{wf} - P_R - P_{clR}\right) \right]_m . \tag{4.2.1-4}$$

4.2.2 Linear Systems for Pressure Equation

The operator on flux term is self-adjoint, so the linear system based on Equation (4.2.1-1) from the discretization of Equation (4.1.2-1) is a symmetrical matrix. And from the basic property of the difference formulations of the continuity equation, this linear system is also diagonal-dominant and irreducible, thus, the linear system is positive definite (Gupta et al. 1986). That is,

$$Au = RHS , \tag{4.2.2-1}$$

where matrix A has a heptagonal sparse pattern as shown in Figure 4.2.2-1. However, matrix A is a symmetric, positive and definite (SPD) matrix, and its gridblock numbering takes the X-direction first, Y-direction second, and Z-direction last. Figure 4.2.2-1 shows the different nonzero pattern for an example with 4x4x3 gridblocks compared to UTCOMP in Figure 3.2.2-1.

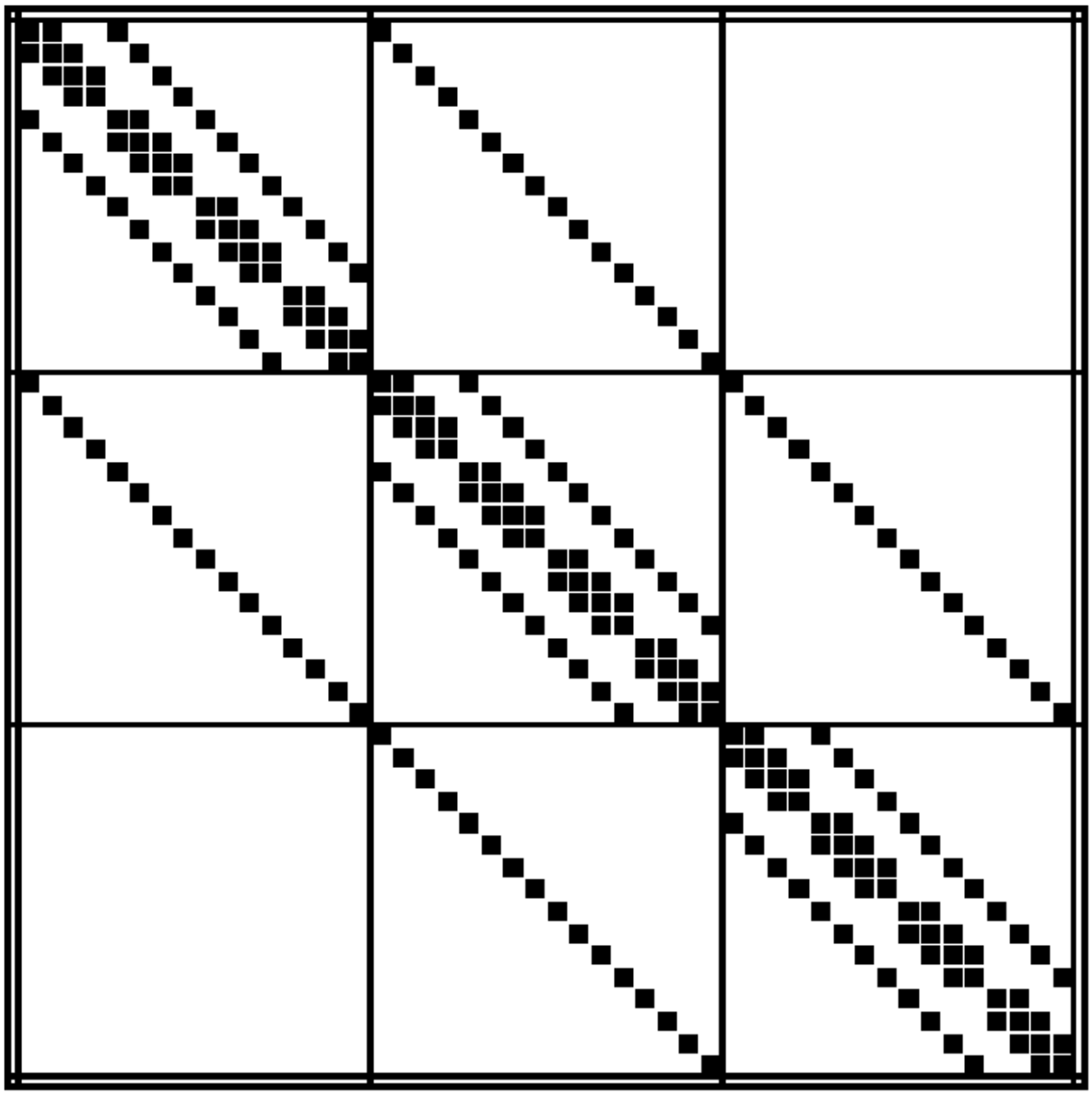

Figure 4.2.2-1: Heptagonal sparse pattern for 4x4x3 gridblocks from UTCHEM.

4.3 Linear Solvers in UTCHEM

4.3.1 Iterative Method and Preconditioner in UTCHEM

The existing solver in UTCHEM is CG iterative method plus Jacobi preconiditioner (JCG). The flowchart of JCG used in UTCHEM is presented in Figure 4.4-1.

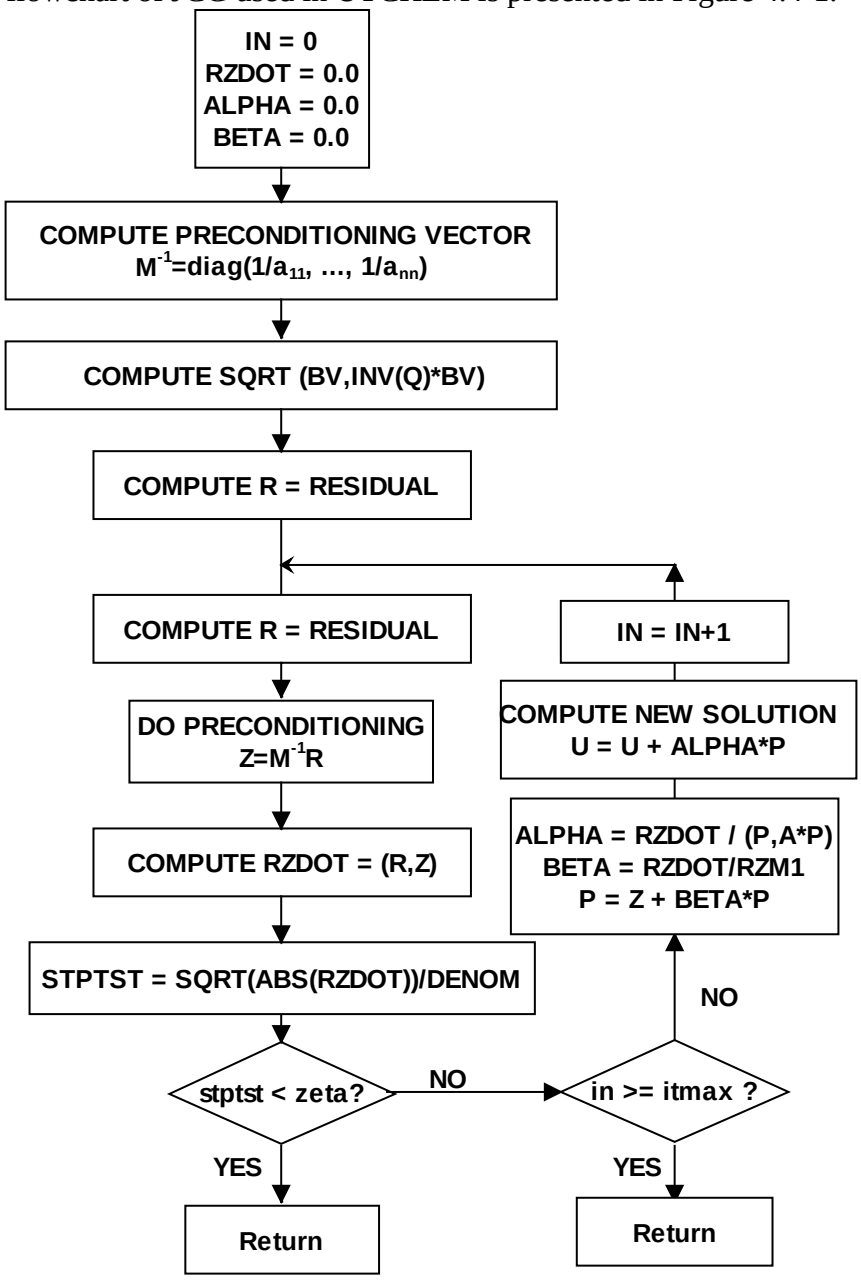

Figure 4.4-1: JCG's flowchart used in UTCHEM.

4. 4 Implementation of Both BoomerAMG and SAMG in UTCHEM

In this section, we will show the implementation of BoomerAMG and SAMG in UTCHEM.

4.4.1 Options of Both BoomerAMG and SAMG in Flowchart

First of all, we show the place of new options of BoomerAMG and SAMG in the flowchart of UTCHEM. Figure 4.5-1 shows that the location of SAMG and BoomerAMG was added into UTCHEM.

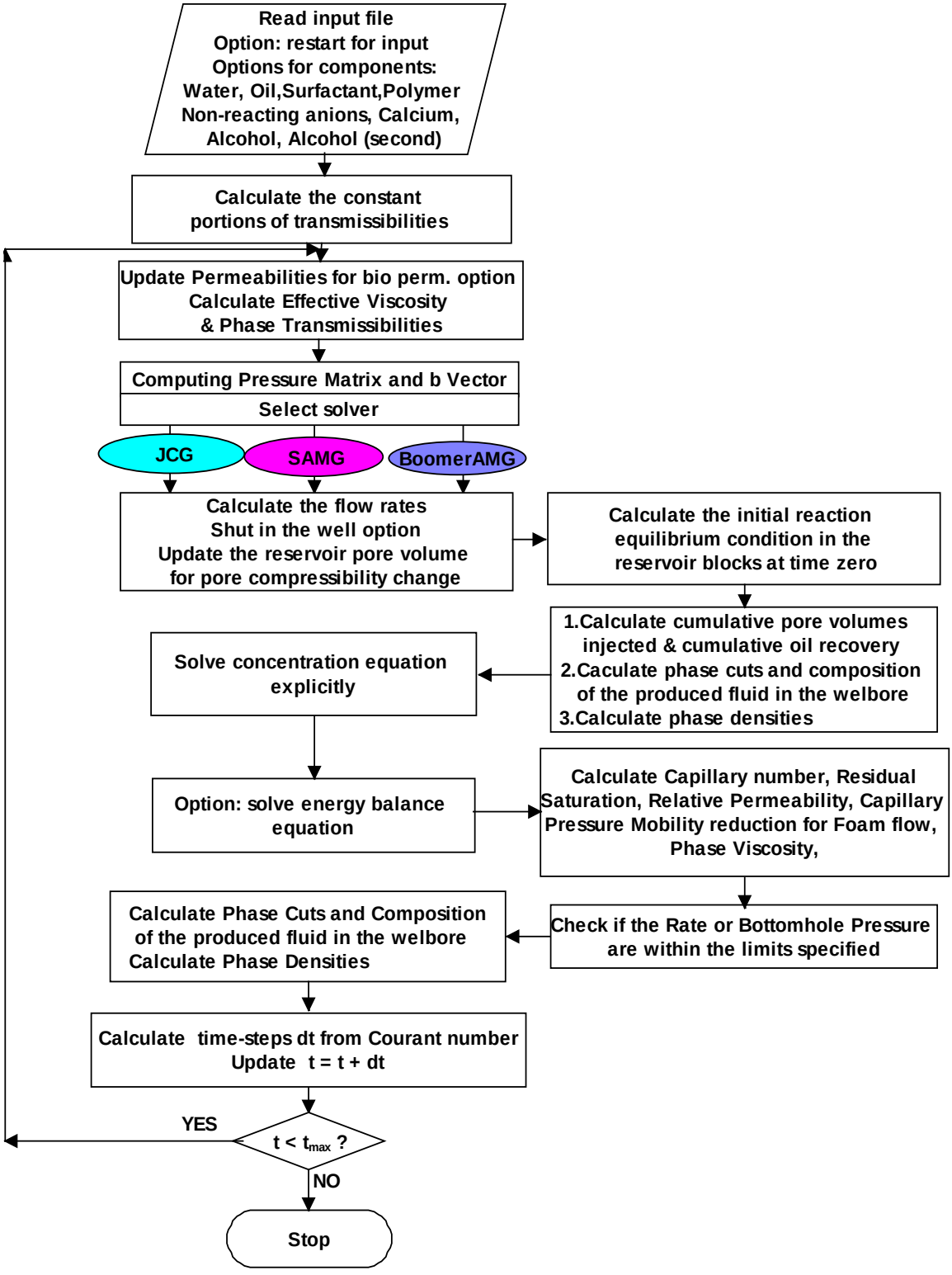

Figure 4.4.1-1: Flowchart of UTCHEM with new solvers.

4.4.2 Implementation of Both Solvers

The first step of the implementation is defining an index to choose a different solver using different values. “ITCK” flag was defined as a global integer variable. Using this flag, the user can choose the solver he/she wants. “ITCK” flag was read in main program file “aamain.f” from an input file “solver.itck” in Figure 4.4.2-1.

```
open(66,file='solver.itck')
READ(66,*)
READ(66,*) ITCK
close(66)
```

Figure 4.4.2-1: Input “ITCK” flag for solver option in “aamain.f” in UTCHEM.

The second step is calling a solver when the index was assigned the solver’s option. In “solmat.f” file, we used “ITCK” flag to add options for the new solver (Figure 4.4.2-2)

```
ZETAL = TOL
ITERL = MITER
IF(3.EQ.ITCK) THEN
   WRITE(*,*)'    HYPRE    IS  WORKING '
   CALL CPU_TIME(TOLD)
   CALL HYPRE0 (ZETAL,ITERL,IER)
   CALL CPU_TIME(TNEW)
ELSE IF(2.EQ.ITCK) THEN
   WRITE(*,*)'    SAMG IS WORKING '
   CALL CPU_TIME(TOLD)
   CALL SAMG0 (ZETAL,ITERL,IER)
   CALL CPU_TIME(TNEW)
ELSE
   WRITE(*,*)'    JCG IS WORKING '
   CALL CPU_TIME(TOLD)
   CALL JCG (ZETAL,ITERL,IER)
   CALL CPU_TIME(TOLD)
ENDIF
```

Figure 4.4.2-2: options for new solvers in “solmat.f” in UTCHEM.

There are three steps for connecting SAMG and BoomerAMG with UTCHEM as follows in Figure 4.4.2-3.

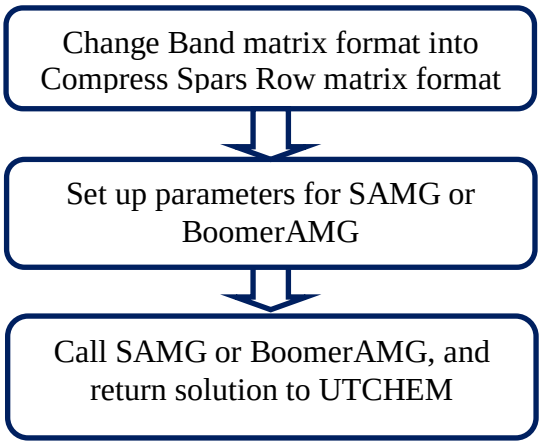

Figure 4.4.2-3: Connecting SAMG and BoomerAMG with UTCHEM.

4.5 Numerical Results

There are three solvers in UTCHEM, JCG, SAMG, and BoomerAMG, that have been implemented into the simulator. We will compare them using the following test problems.

4.5.1 Water Injection (Case I)

A water injection test was performed first. The linear solver's tolerance was set to 10^{-12}. Reservoir properties for this case are displayed in Table 4.5.1-1.

Table 4.5.1-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Formation compressibility	0.0
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.4
Initial reservoir pressure	3100 psi
Permeability in the X-direction	500 md
Permeability in the Y-direction	100 md
Permeability in the Z-direction	50 md
Number of gridblocks (NX*NY*NZ)	150x150x2
Porosity	0.2
Grid block size	22.727×22.727×5 ft^3
Number of components	2
Component's names	Water, Oil
Number of wells	2
Type of wells	Injection well, production well
Simulation time	100 days
Maximal injection rate	1000 feet^3/day
Maximal BHP of injector	5000 psi
Injection contents	100% water
Maximal production rate	5000 feet^3/day
Bottomhole production pressure	3100 psi

Figure 4.5.1-1 shows the reservoir region and wells for Case I. The injection well injected 100% water with the maximum injection rate at 1000 ft^3/day. The production well was kept at a constant bottomhole pressure of 3,100 psi.

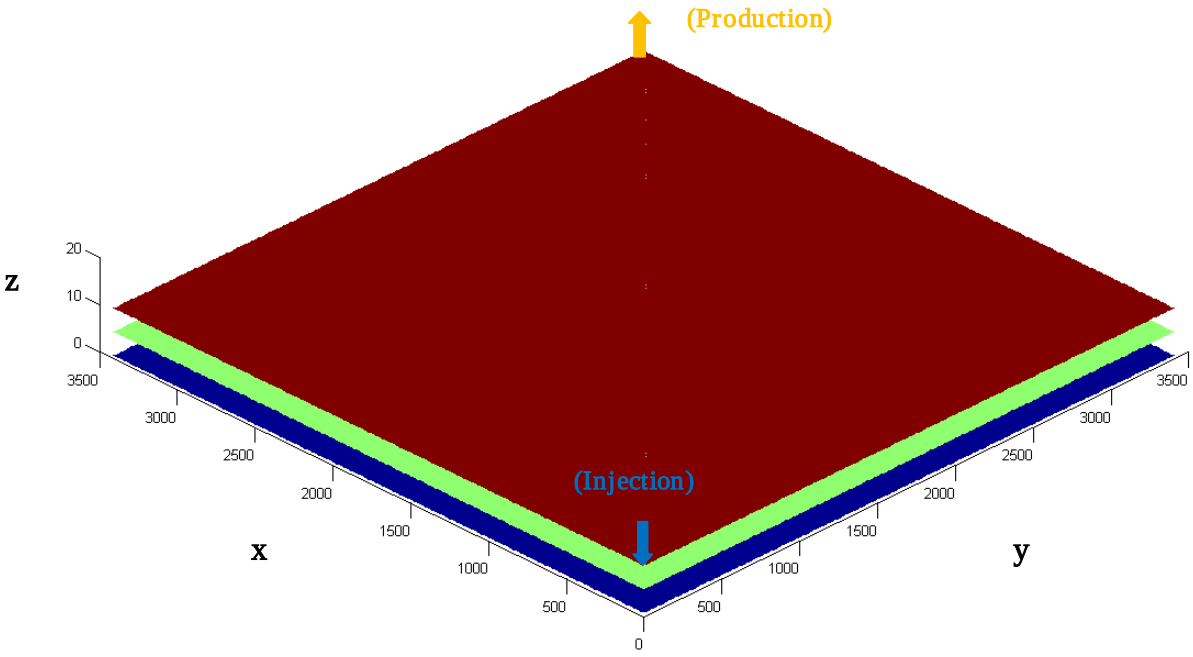

Figure 4.5.1-1: Reservoir model for Case I.

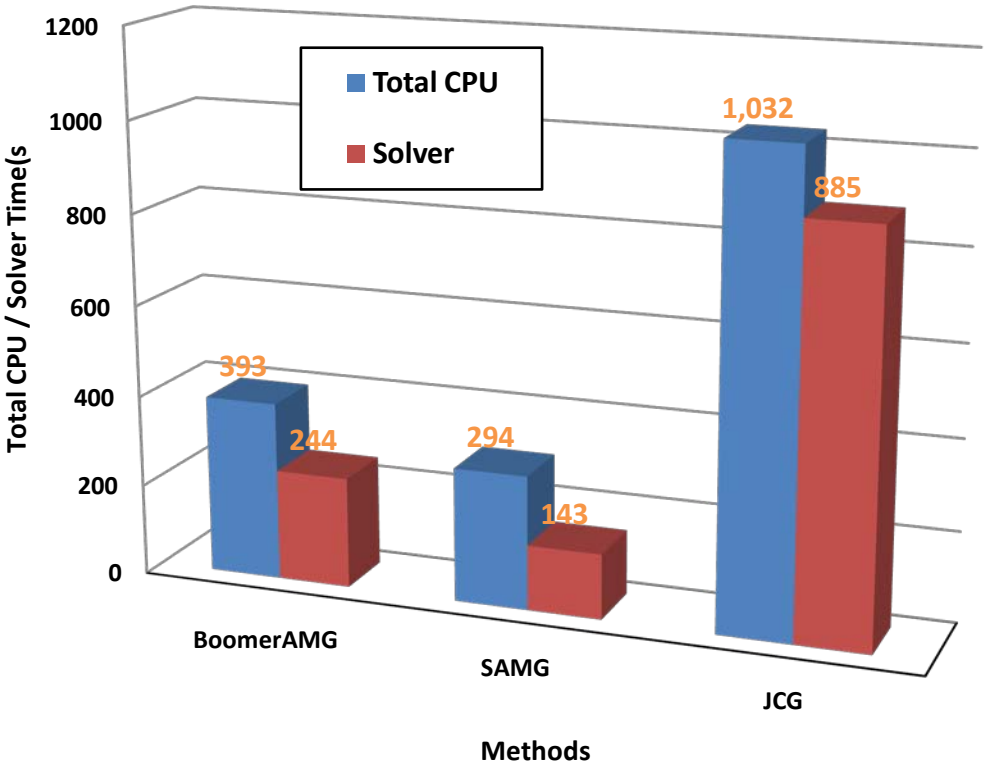

Figure 4.5.1-2: Comparison of running times for Case I.

Figure 4.5.1-2 shows solver and CPU times in seconds spent for Case I using the three solvers, BoomerAMG, SAMG, and JCG. The running times show that SAMG is the fastest solver, while BoomerAMG is the second fastest. SAMG took less than 58% of the running time of BoomerAMG, and 16% of JCG's running time.

4.5.2 Surfactant Polymer (SP) Flooding (Case II)

SP flooding test was conducted second. The linear solver's tolerance was set to 10^{-12}. Reservoir properties and simulation parameters are displayed in Table 4.5.2-1.

Table 4.5.2-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Number of gridblocks (NX*NY*NZ)	110x110x10
Grid block size	22.727×22.727×5 ft^3
Number of components	5
Component	Water, Oil, Surfactant, Polymer, Anion
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	150 days
Formation compressibility	0.0
Oil compressibility	0.0000025
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.65
Initial reservoir pressure	3100 psi
Maximal injection rate	10,000 $feet^3$/day
Maximum BHP of injection	100,000 psi
Injection fluid composition	98% water, 2% Surfactant (500 ppm Polymer, 0.15 meq/ml Anion)
Permeability in the X,Y-direction	500;400;300;200;150;140;130;120;110;100md
Permeability in the Z-direction	0.1 md
Max Bottomhole production pressure	5,000 psi
Max rate production pressure	1,000 ft^3/day
Porosity	0.2

Figure 4.5.2-1 shows the reservoir region and wells for Case II. The injection well injected fluid consisted of 2% surfactant, and 500 ppm polymer with a maximum injection rate of 10,000 ft^3/day.

The heterogeneity of permeability in the X-direction is shown in Figure 4.5.2-2, in which different layers have a different permeability. The permeability decreases from the top layer to the bottom layer.

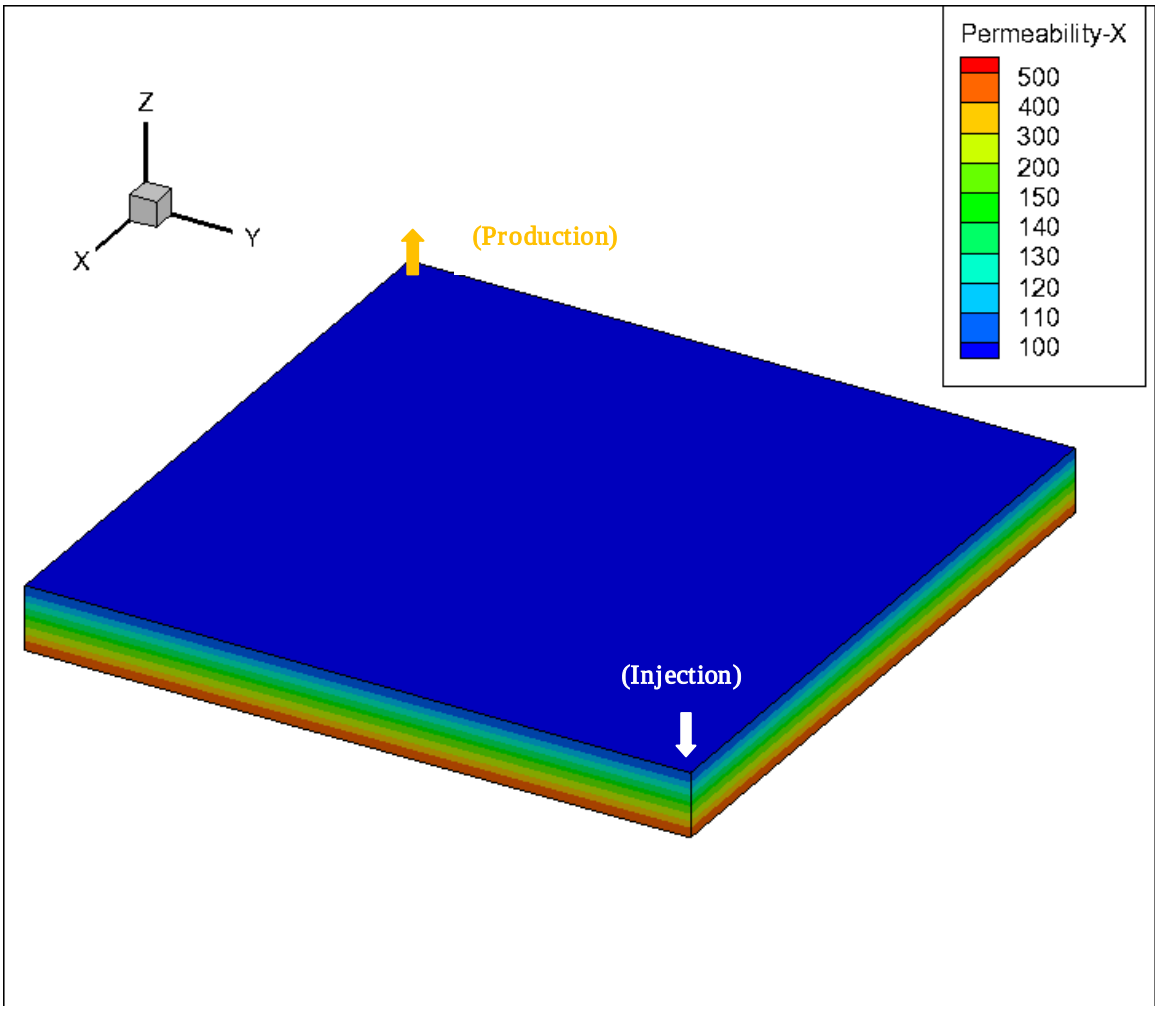

Figure 4.5.2-2: The heterogeneity of permeability in the X-direction for Case II.

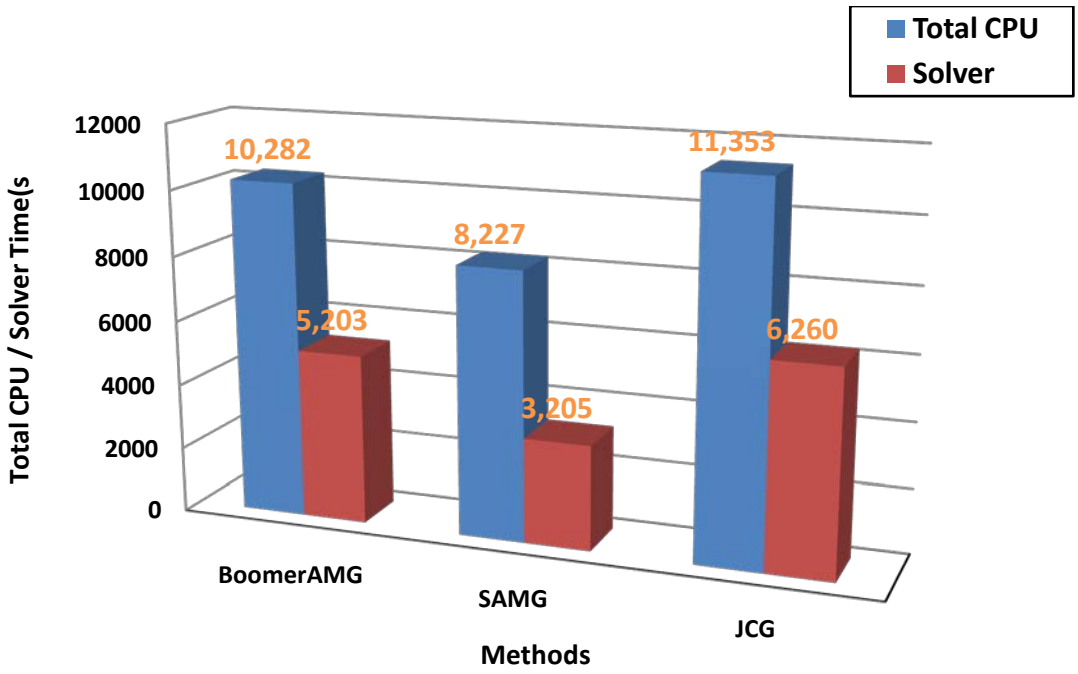

Figure 4.5.2-2: Comparison of running times for Case II.

Figure 4.5.2 shows the CPU times for Case II using UTCHEM with three linear solvers, BoomerAMG, SAMG, and JCG. Figure 4.5.2 shows that SAMG is the fastest solver, which is more than 1.6 times faster than BoomerAMG, and more than 1.9 times faster than JCG. BoomerAMG is the second fastest and is 1.2 times faster than JCG.

4.5.3 Surfactant Polymer Flooding (Case III)

The test of surfactant polymer flooding for a six-component reservoir was conducted third. The linear solver's tolerance was set to 10^{-12}. Reservoir properties and simulation parameters are displayed in Table 4.5.3-1.

Table 4.5.3-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Formation compressibility	0.0
Fluid compressibility	0.0
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.60
Initial reservoir pressure	3100 psi
Permeability in the X-direction	See Figure 4.5.3-1
Permeability in the Y-direction	See Figure 4.5.3-1
Permeability in the Z-direction	50 md
Porosity	0.2
Number of gridblocks (NX*NY*NZ)	100×100×20
Grid block size	22.727×22.727×5 ft^3
Number of components	6
Component	Water,Oil,Surfactant,Polymer, Anion,Calcium
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	500 days
Maximum injection rate	1,000 ft^3/day
Maximum BHP of injection	2,000 psi
Injection contents	98% water, 2% Surfactant (500 ppm Polymer and 0.15 meqml Anion)
Maximum production rate	10,000 feet^3/day
Max Bottomhole production pressure	5,000 psi
Max rate production pressure	1,000 feet^3/day

The injection well injects fluid consisting of 98% water, 2% surfactant, 500 ppm polymer with a maximum injection bottomhole pressure of 10,000 psi and a maximum injection rate of 10,000 ft^3/day. The production well was kept at a bottomhole pressure of 5,000 psi and a maximum production rate of 1,000 ft^3/day.

The heterogeneity of permeability in the X-direction and the Y-direction is shown in Figure 4.5.3-1, in which the top ten layers in blue color have a constant permeability in blue color, and the bottom ten layers in green color have another constant permeability in green color; the permeability decreases from the top ten layers to the bottom ten layers.

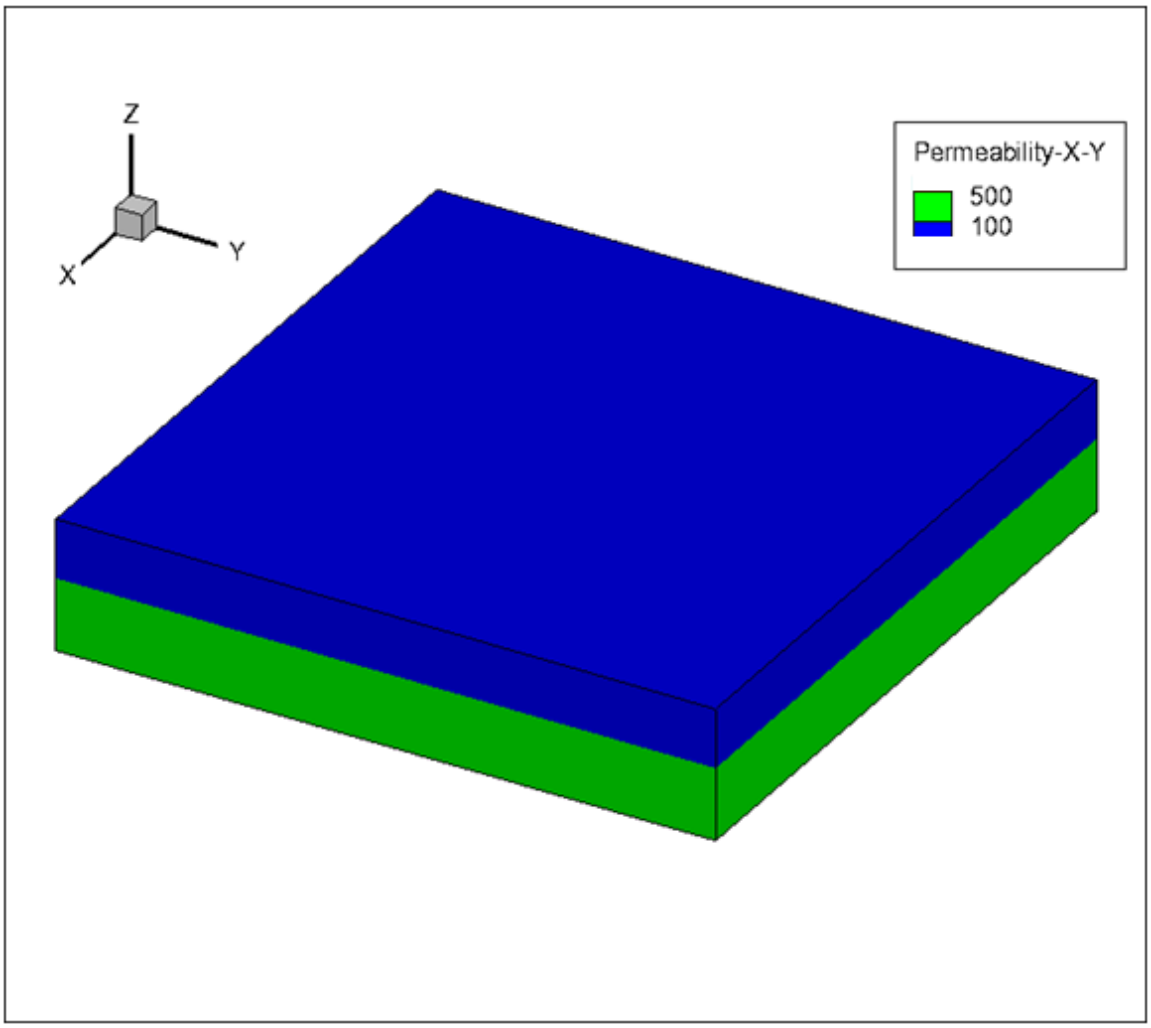

Figure 4.5.3-1: The heterogeneity of permeability in the X-direction for Case III.

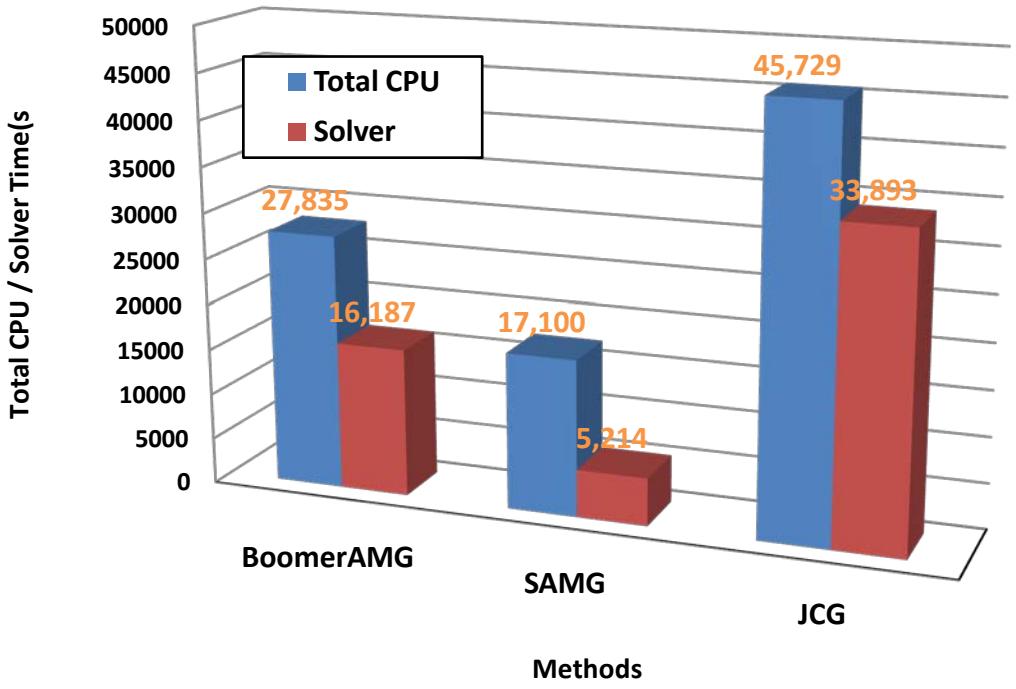

Figure 4.5.3-2: Comparison of running times for Case III.

Figure 4.5.3-2 shows the computational time for Test 4.5.3 using UTCHEM for three linear solvers, BoomerAMG, SAMG, and JCG. Figure 4.5.3-2 shows that JCG is much slower than SAMG and BoomerAMG. SAMG is the fastest solver among the three solvers for this case, which is less than 32% of the running time of BoomerAMG, and less than 15% of the running time of JCG.

4.5.4 SP Flooding for Heterogeneous Permeability in the X-Direction (Case IV)

The test of surfactant polymer flooding for a six-component reservoir is presented in this section. The linear solver's tolerance was set to 10^{-12}. Reservoir properties and simulation parameter are displayed in Table 4.5.4-1.

Table 4.5.4-1: Reservoir properties of Case IV.

Parameter	**Value**
Formation compressibility	0.0
Fluid compressibility	0.0
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.60
Initial reservoir pressure	3100 psi
Permeability in the X-direction	Heterogeneous(Figure 4.5.4-2)
Permeability in the Y-direction	500;100;500;100;500;100;500;100md(Figure 4.5.4-3)
Permeability in the Z-direction	50 md
Porosity	0.2
Number of gridblocks (NX*NY*NZ)	64×64×8
Gridblock size	20×20×5 ft^3
Number of components	6
Component's names	Water,Oil,Surfactant,Polymer, Anion,Calcium
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	500 days
Maximum of injection rate	10,000 feet^3/day
Maximum of BHP of injection	5,000 psi
Injection contents	97%water, 3% Surfactant(1500 ppm Polymer, 0.17 meq/ml Anion, 0.01 meq/ml calcium)
Max bottomhole production pressure	5,000 psi
Max rate production	50,000feet^3/day

The injection well injected fluid consisting of 97% water, 3% surfactant, (1500 ppm Polymer, 0.17 meq/ml Anion, 0.01 meq/ml calcium for aqueous phase) with a maximum injection bottomhole pressure of 5,000 psi and a maximum injection rate of 1,000 ft^3/day. The production well was kept at a maximum bottomhole pressure of 5,000 psi and a maximum production rate of 50,000 ft^3/day.

The heterogeneity of permeability in the X-direction is shown in Figure 4.5.4-1, which was constructed with the condition that Dykstra-Parsons coefficients of 0.75.

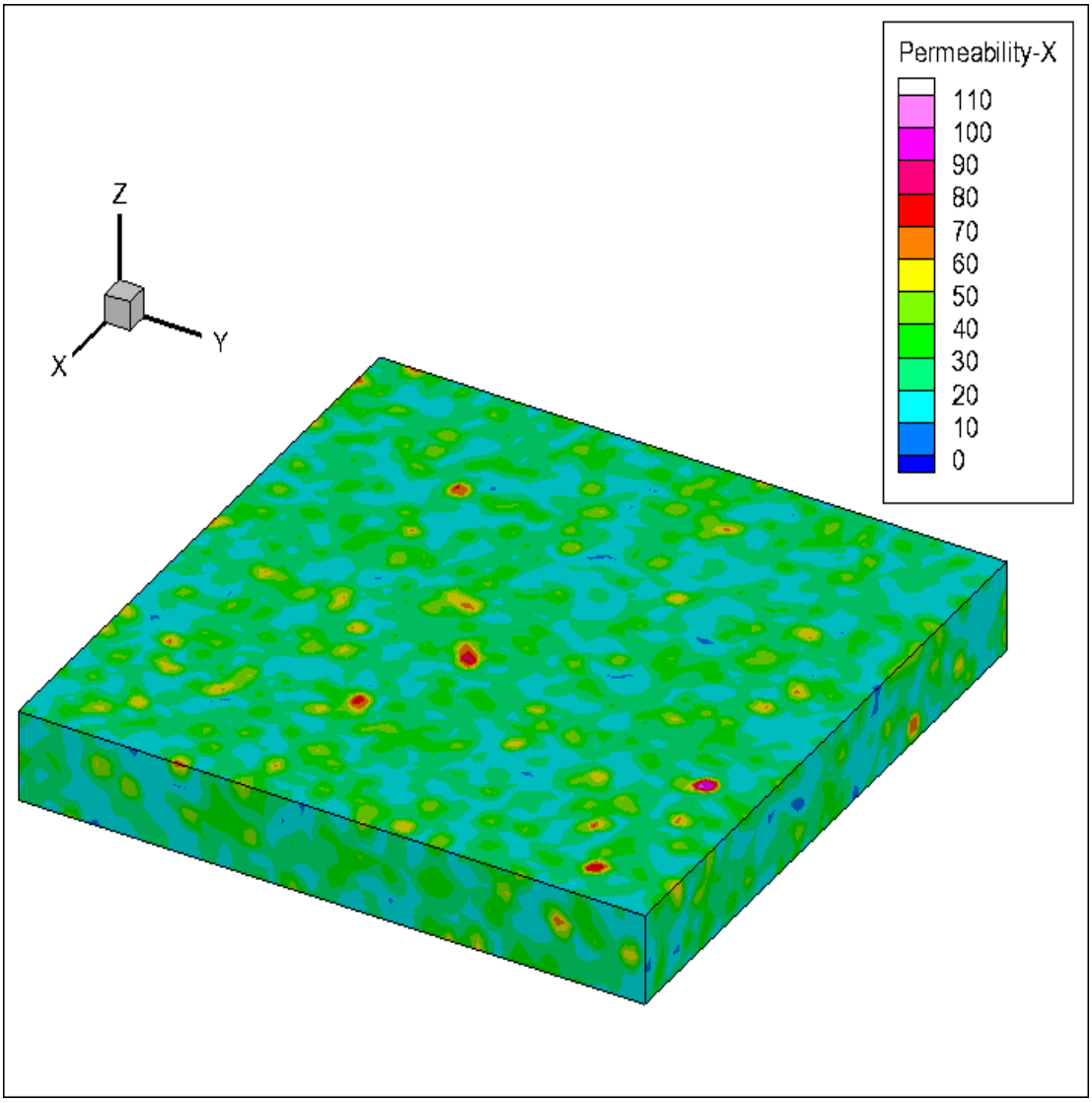

Figure 4.5.4-1: The heterogeneous permeability in the X-direction for Case IV.

The heterogeneity of permeability in the Y-direction is shown in Figure 4.5.4-2, in which each odd or even number of the layer has the same permeability: odd numbered layers' permeability is 500 md and even number layers' permeability is 100 md.

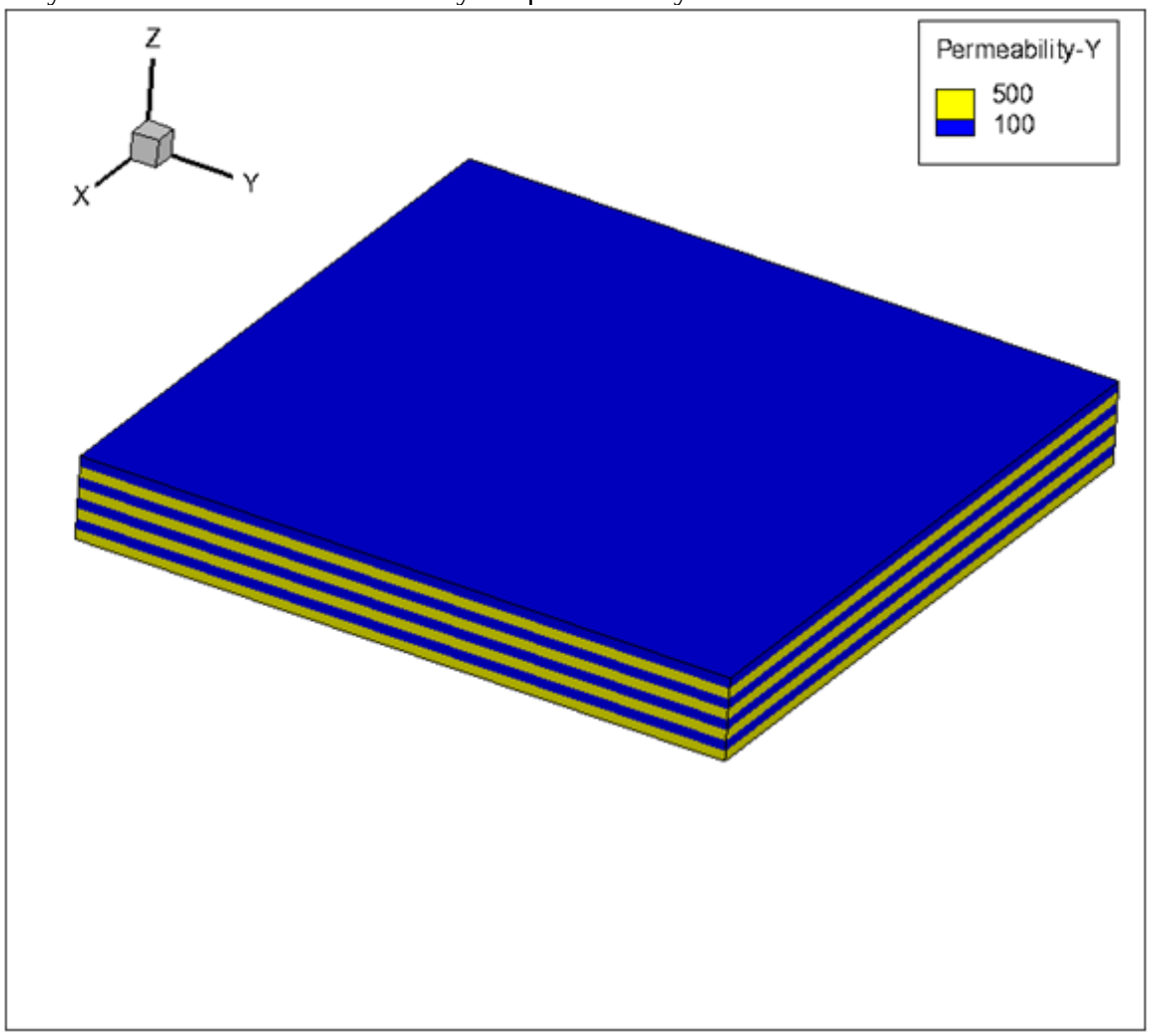

Figure 4.5.4-2: Permeability in the Y-direction for Case IV.

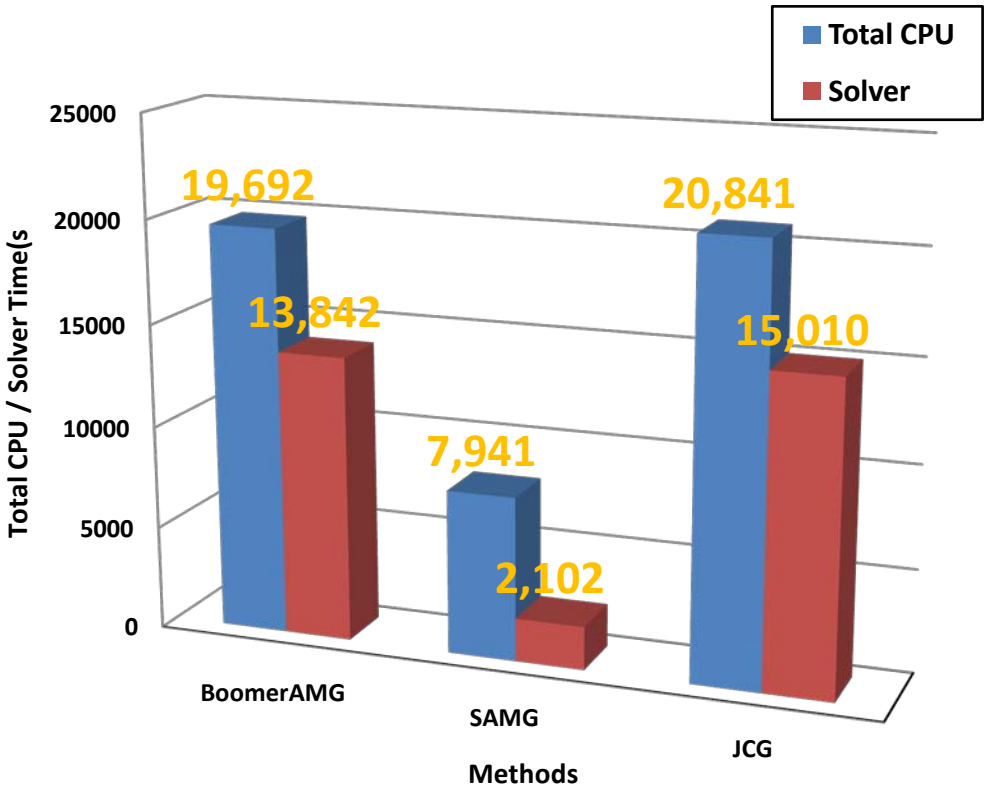

Figure 4.5.4-3: Comparison of running times for Case IV.

Figure 4.5.4-3 shows the CPU times for Case IV using UTCHEM for three linear solvers, BoomerAMG, SAMG, and JCG. The CPU times show JCG is much slower than SAMG and BoomerAMG. SAMG is the fastest solver among all three solvers in this case, which is less than 15% of the running time of BoomerAMG, and less than 14% of the running time of JCG.

4.5.5 SP Flooding for Heterogeneous Permeability in the X, Y, Z-Directions (Case V)

The test of surfactant polymer flooding for a six-component reservoir was conducted next. The tolerance was set to 10^{-12}. Reservoir properties and simulation parameters are displayed in Table 4.5.5-1, where DPC means Dykstra-Parsons coefficient.

Table 4.5.5-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Number of gridblocks (NX*NY*NZ)	64×64×8
Grid block size	20×20×5 ft^3
Number of components	6
Component	Water, Oil, Surfactant, Polymer, Anion, Calcium
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	500 days
Formation compressibility	0.0
Fluid compressibility	0.0
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.60
Initial reservoir pressure	3,100 psi
Maximum of injection rate	1,000 $feet^3$/day
Maximum of BHP of injection	2,000 psi
Injection contents	98% water, 2% Surfactant (500 ppm Polymer 0.15 meq/ml Anion for aqueous phase)
Maximum of production rate	10,000 $feet^3$/day
Maximum of BHP of production	100,000 psi
Permeability in the X-direction	Heterogeneous with DPCs=0.5
Permeability in the Y-direction	Heterogeneous with DPCs=0.7
Permeability in the Z-direction	Heterogeneous with DPCs=0.9
Max Bottomhole production pressure	5,000 psi
Max rate production pressure	50,000 $feet^3$/day
Porosity	0.2

Except for the differences on the permeability, Case V kept the same other reservoir properties evident in Case IV.

The heterogeneity of permeability in the X-direction is shown in Figure 4.5.5-1, which was constructed with the condition that DPCs equal 0.5.

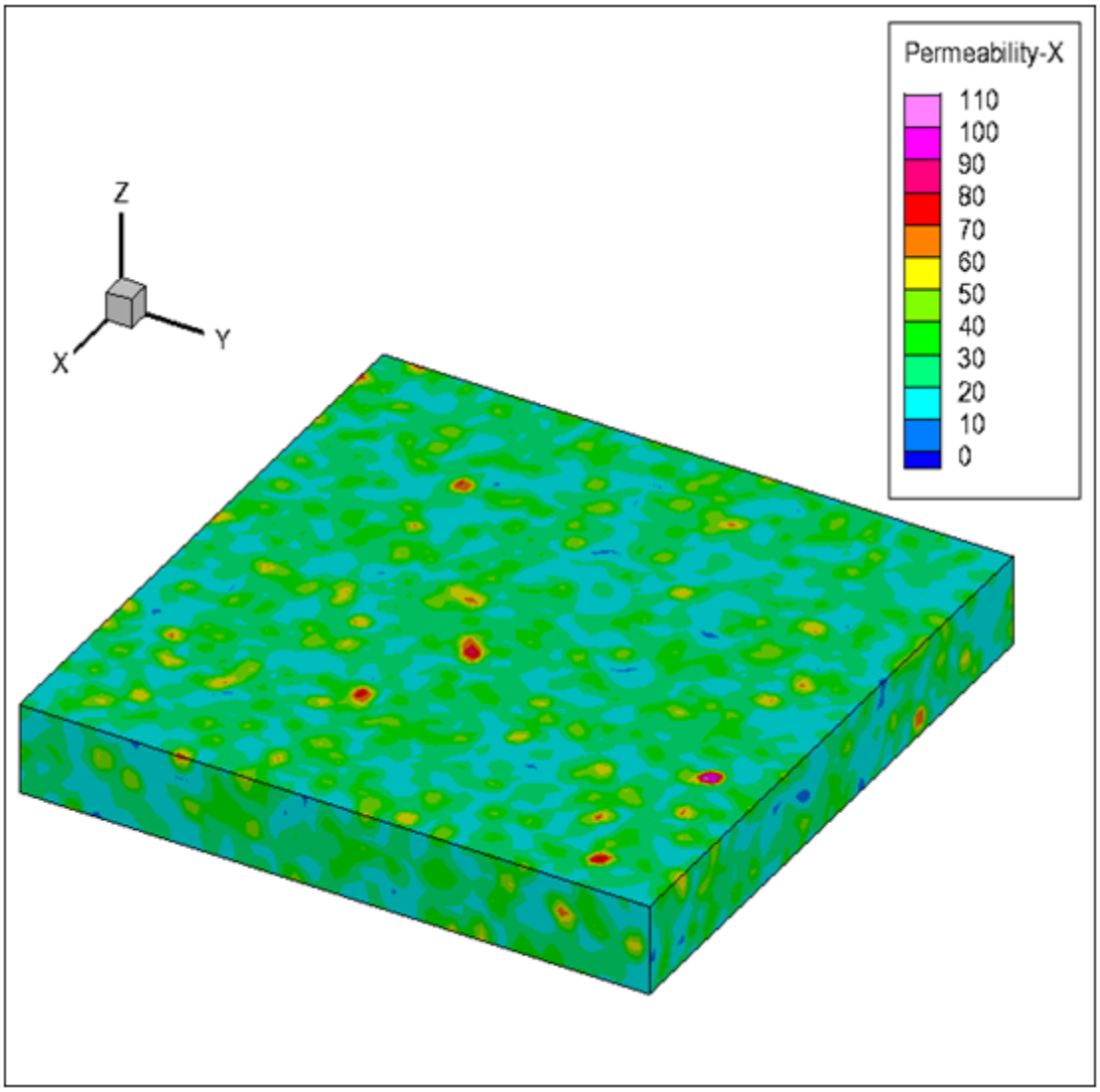

Figure 4.5.5-1: The heterogeneous permeability in the X-direction for Case V.

The heterogeneity of permeability in the Y-direction is shown in Figure 4.5.5-2, which was constructed with the condition that DPCs equal 0.7.

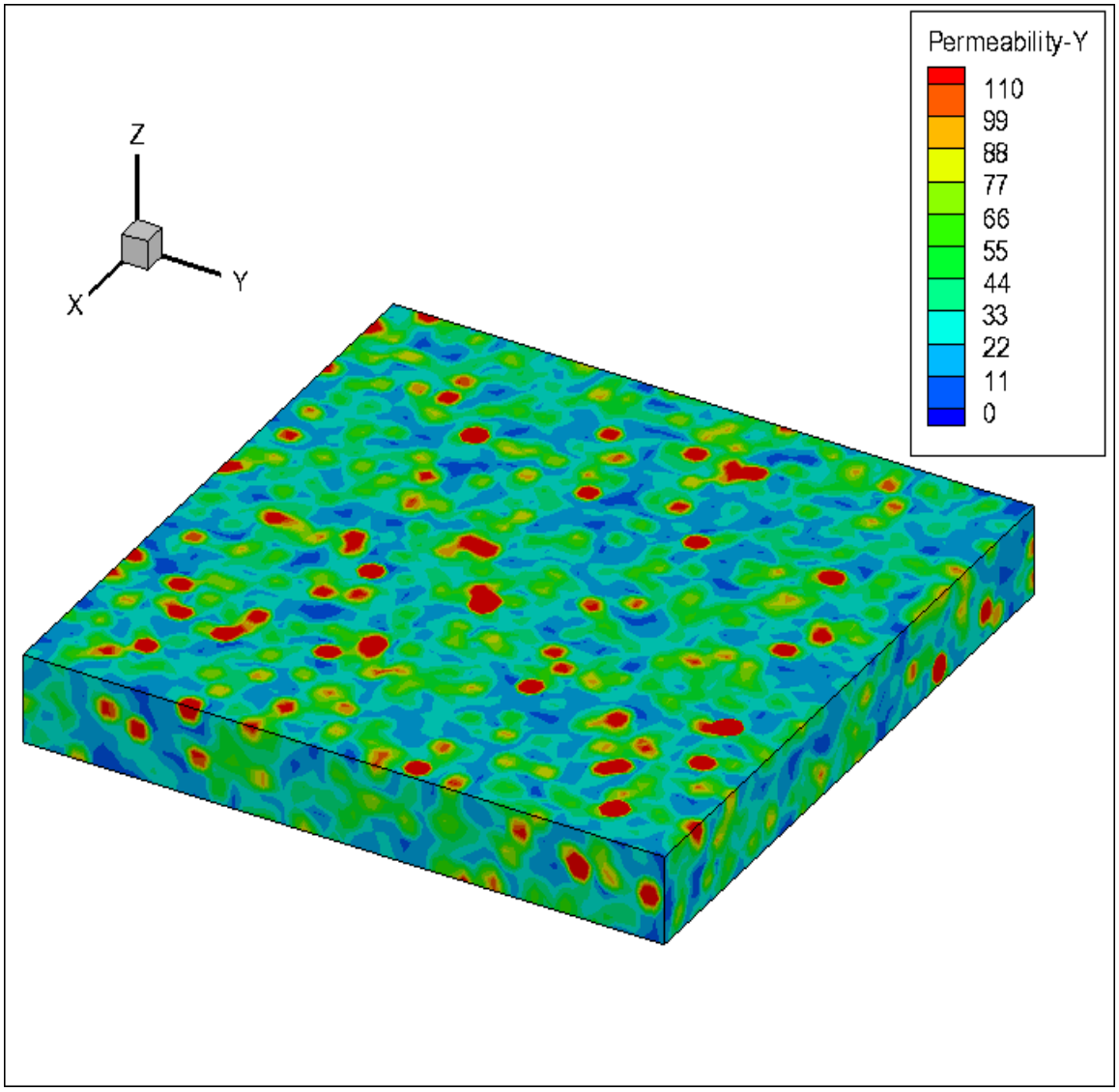

Figure 4.5.5-2: The heterogeneous permeability in the Y-direction for Case V.

The heterogeneity of permeability in the Z-direction is shown in Figure 4.5.5-3, which was constructed with the condition that Dykstra-Parsons coefficients equals 0.9.

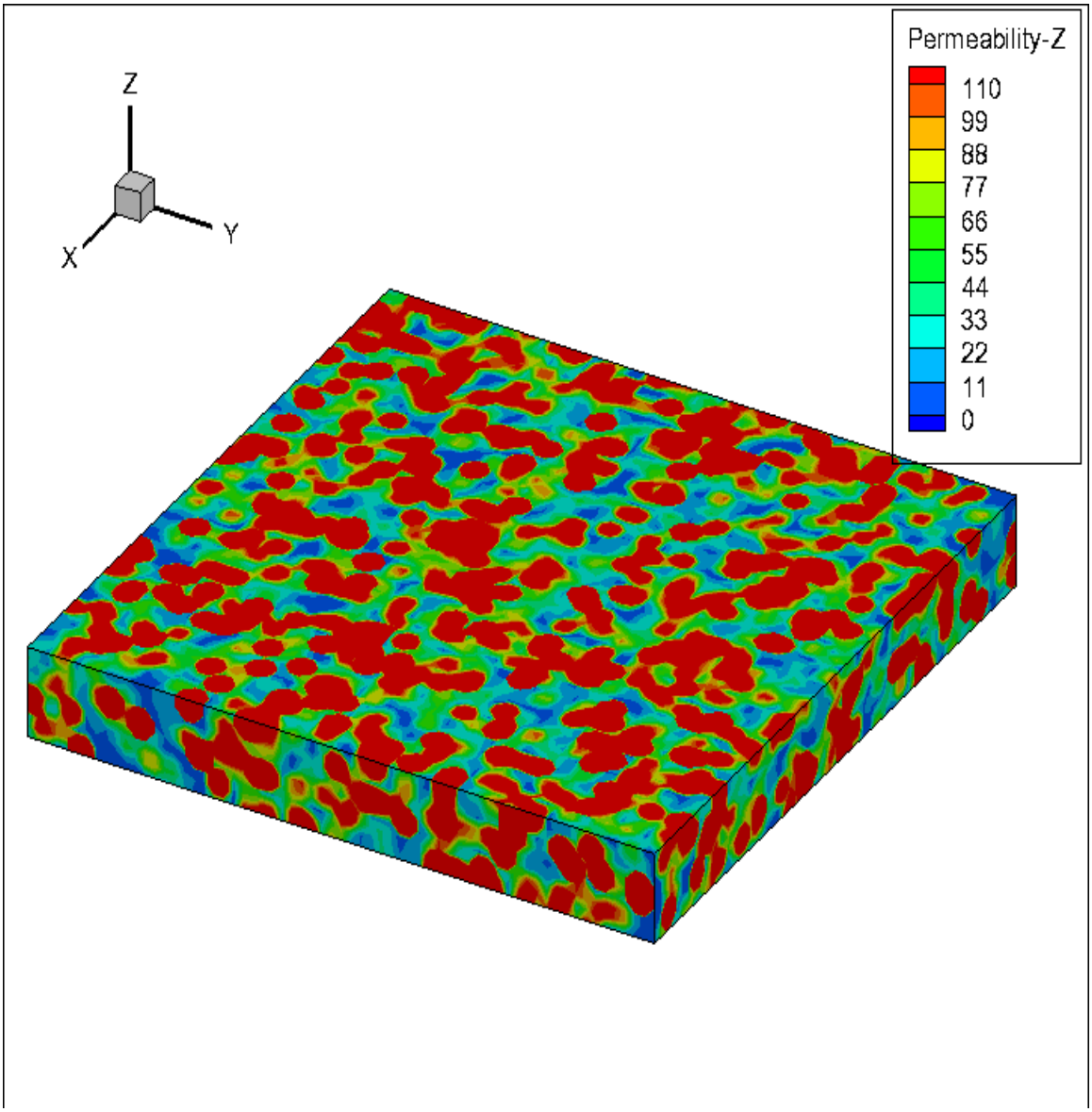

Figure 4.5.5-3: The heterogeneous permeability in the Z-direction for Case V.

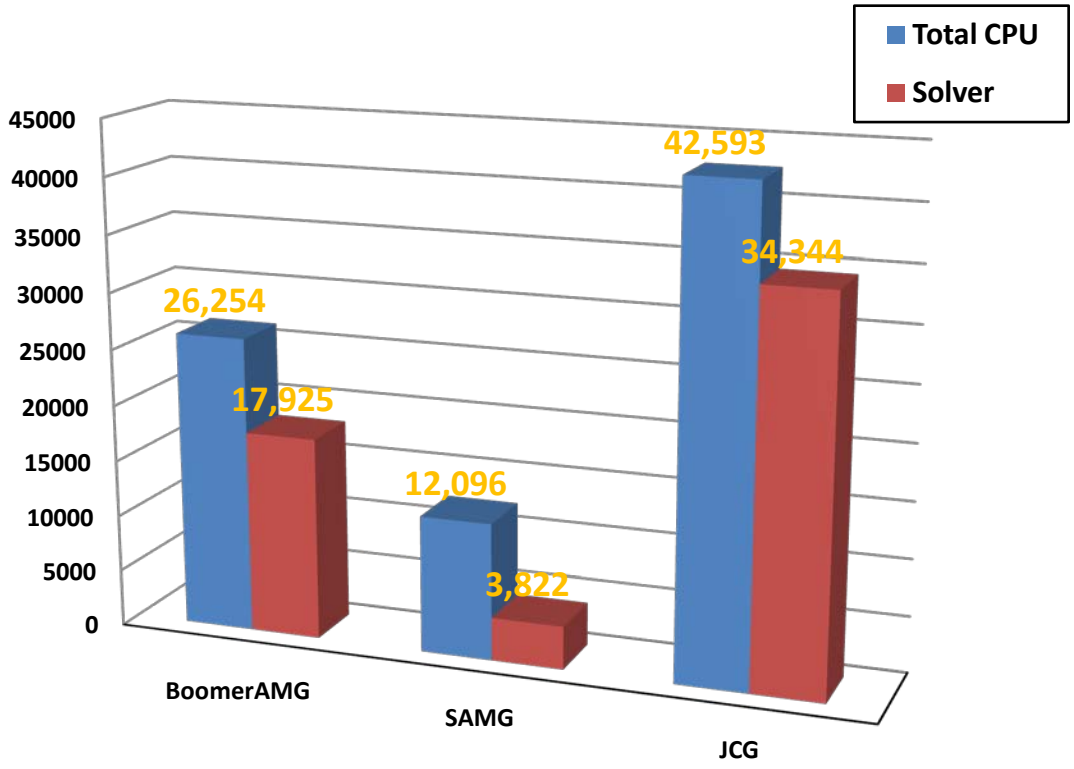

Figure 4.5.5-4: Comparison of running times for Case V.

Figure 4.5.5-4 shows the CPU times for Case V using UTCHEM for three linear solvers, BoomerAMG, SAMG, and JCG. The CPU times show JCG is much slower than SAMG and BoomerAMG. SAMG is the fastest solver among all three solvers in this case, which is less than 21% of the running time of BoomerAMG, and less than 11% of the running time of JCG.

4.5.6 Comments and Conclusions

Section 4.5 includes five tests divided into two types: both the homogeneous and the heterogeneous permeability tests.

The first group tests are the homogenous-permeability tests. The numerical results show BoomerAMG is much faster than JCG, where, for the best case, BoomerAMG took only one-third of JCG's running time. In other words, AMG is much faster than a Krylov subspace iterative method. However, SAMG is much more efficient than BoomerAMG because SAMG includes some efficient smooth operators. Also, interpolation operators remove errors quicker than BoomerAMG. SAMG only took less than one-third of the running time of BoomerAMG in its best run case.

Case IV and Case V are for the heterogeneous-permeability reservoir case studies. The permeability in the X-direction of Case IV was generated with DPCs equal to 0.5. The permeability in the Y-direction of Case IV is constant for each layer on the X-Y plane. The permeability in the Z-direction of Case IV is also constant. BoomerAMG is about 5% faster than JCG, and SAMG took less than one-sixth of the running time of BoomerAMG. Case V increased the heterogeneity of Case IV in the Y-direction and the Z-direction. Not only were the permeabilities in the X-direction generated with DPCs equal to 0.5, but the permeabilities in the X-direction and the Y-direction also were generated with DPCs equaling 0.7 and 0.9 in Case V. The three linear solvers in Case V increased the computational times compared to Case IV. The biggest increase in computational time was from JCG, which was 1.27 times of the JCG's running time of Case IV. The second biggest increase in computational time was from SAMG, which was 80% of the SAMG's running time in Case IV. BoomerAMG had the smallest increase, about 30% of BoomerAMG's running time. When comparing the three methods in Case V, we can see that BoomerAMG took about half of running time of JCG, and

SAMG took one-fourth of the running time of BoomerAMG. We may draw the conclusion that SAMG is much more efficient for solving the heterogeneous-permeability problems than BoomerAMG, and BoomerAMG is much more efficient for solving the heterogeneous-permeability problems than JCG.

Chapter 5: GPAS and Linear Solvers

GPAS is a fully implicit simulator that can be used on single as well as multiprocessor computers. GPAS contains both an EOS-based compositional simulation model that is similar to that of UTCOMP and a chemical flooding model that is similar to that of UTCHEM (Wang et al. 1997, John et al. 2004). These two models will be discussed in this chapter.

5.1 Mathematical Model Equations in GPAS

5.1.1 Governing Equations of Compositional Model in GPAS

Governing equations of compositional simulation in GPAS was first presented by Wang et al. (1997) as follows:

$$\begin{cases} \textbf{Component material balance} (i = 1, \ldots, n_c) \\ V_b \dfrac{\partial}{\partial t}\left(\varphi \sum_{j=1}^{n_p} \left(\xi_j x_{ij} S_j \right) \right) - V_b \nabla \cdot \sum_{j=1}^{n_p} \dfrac{\overline{\overline{k}} k_{rj}}{\mu_j} \xi_j x_{ij} \left(\nabla P_j - \gamma_j \nabla D \right) - q_i = 0 \\ \\ \textbf{Phase} - \textbf{equilibrium relationship} \\ \ln\left(f_i^{\,j} \right) - \ln\left(f_i^{\,r} \right) = 0 \\ \\ \textbf{Volume and capillary constraints} \\ V_b \sum_{i=1}^{n_c} N_i \sum_{j=1}^{n_p} L_j \bar{v}_i - V_p = 0 \\ \\ \textbf{Molar energy balance} \\ V_b \dfrac{\partial U^T}{\partial t} + V_b \nabla \cdot \sum_{j=a,v,o} \zeta_j h_j \vec{v} - V_b \nabla \cdot (\lambda_T \nabla T) = -\dot{Q}_L + \dot{q}_H \,. \end{cases} \qquad (5.1.1\text{-}1)$$

5.1.2 Governing Equations of Chemical Flooding Model in GPAS

Governing equations of the chemical flooding model in GPAS was extended from the EOS compositional module in Section 5.1.1 with the assumption that chemical species such as tracer, polymer, surfactant, and electrolytes occupy negligible volume and do not affect the EOS model as shown in Equation (5.1.2-1) (John et al. 2004, Fathi-Najafabadi et al. 2009),

$$
\begin{cases}
\textbf{Material balance equation for any hydrocarbon component} \\
(i = 1, ..., n_c) \\
V_b \dfrac{\partial}{\partial t}\left(\varphi N_i\right) - V_b \nabla \cdot \dfrac{\overline{\overline{k}}k_{r,ole}}{\mu_{ole}} \rho_{ole} w_{i,ole} \left(\nabla P_{ole} - \gamma_{ole} \nabla D\right) \\
+ V_b \nabla \cdot \dfrac{\overline{\overline{k}}k_{rME}}{\mu_{ME}} \rho_{ME} w_{iME} \left(\nabla P_{ME} - \gamma_{ME} \nabla D\right) = q_i \\
\\
\textbf{Material balance equation for any aqueous component} \\
(i = n_c + 1, ..., n_c + n_a + 1) \\
V_b \dfrac{\partial}{\partial t}\left(\varphi N_i\right) - V_b \nabla \cdot \dfrac{\overline{\overline{k}}k_{r,aqu}}{\mu_{aqu}} \rho_{aqu} w_{i,aqu} \left(\nabla P_{aqu} - \gamma_{aqu} \nabla D\right) \\
+ V_b \nabla \cdot \dfrac{\overline{\overline{k}}k_{rME}}{\mu_{ME}} \rho_{ME} w_{iME} \left(\nabla P_{ME} - \gamma_{ME} \nabla D\right) = q_i \\
\\
\textbf{Volumetric constraint equation} \\
R_V = \dfrac{N_{water}}{\rho_{water}} + \dfrac{N_{surf}}{\rho_{surf}} + \dfrac{1}{\rho_o} \sum_{i=1}^{n_c} N_i - 1 = 0 .
\end{cases}
\tag{5.1.2-1}
$$

5.2 Linear Systems of GPAS

5.2.1 Discretization of Model Equations

Both EOS compositional modules in Section 5.1.1 and Chemical models in Section 5.1.2 are discretized using a fully implicit approximation with single point upstream weighting for transmissibility terms, then these non-linear discretized equations generated a Newton iteration for the solution procedure, in which a linear system generated by computing derivatives of the governing equations is solved to provide updated increments of all unknowns for Newton iterations.

There are two types of reservoir models in GPAS Version 3.5. One reservoir model is the compositional model, which includes $n_p n_c + 2$ unknowns. GPAS has an isothermal compositional model with $n_p = 2.$ The unknowns are: $\mathrm{lnK}_1,\cdots,\mathrm{lnK}_{\mathrm{n_c}}$, $\mathrm{N}_1,\cdots,\mathrm{N}_{\mathrm{n_c}},\mathrm{P}_{\mathrm{aq}}, N_w$. Another reservoir model of GPAS is the chemical flooding model, which includes $n_p(n_c + n_a) + 2$ unknowns. There are phases such as oil phase and aqueous phase. Besides the unknowns in the compositional model, there are $2n_a$ unknowns for the concentrations of aqueous components.

5.2.2 The Properties of Linear Systems

The linear system of equations representing the discretized form of the model equations in GPAS can be obtained from (5.1.1-1) together with the calculation of fluid mass and molar density and volume, viscosity, capillary pressure, relative permeability, and well models as follows:

$$Au = RHS\,, \tag{5.2.2-1}$$

where A has a seven-block-diagonal sparse pattern. Figure 5.2.2-1 shows the spars pattern of A for an example of 4x4x3 gridblocks with one component. There are 3x3 big

sub-block matrices for each X-Y plane layer in it. Figure 5.2.2-2 shows a nonzero pattern for one layer in Figure 5.2.2-1. There are four unknowns in each gridblock marked out by the smallest red squares in Figure 5.2.2-2.

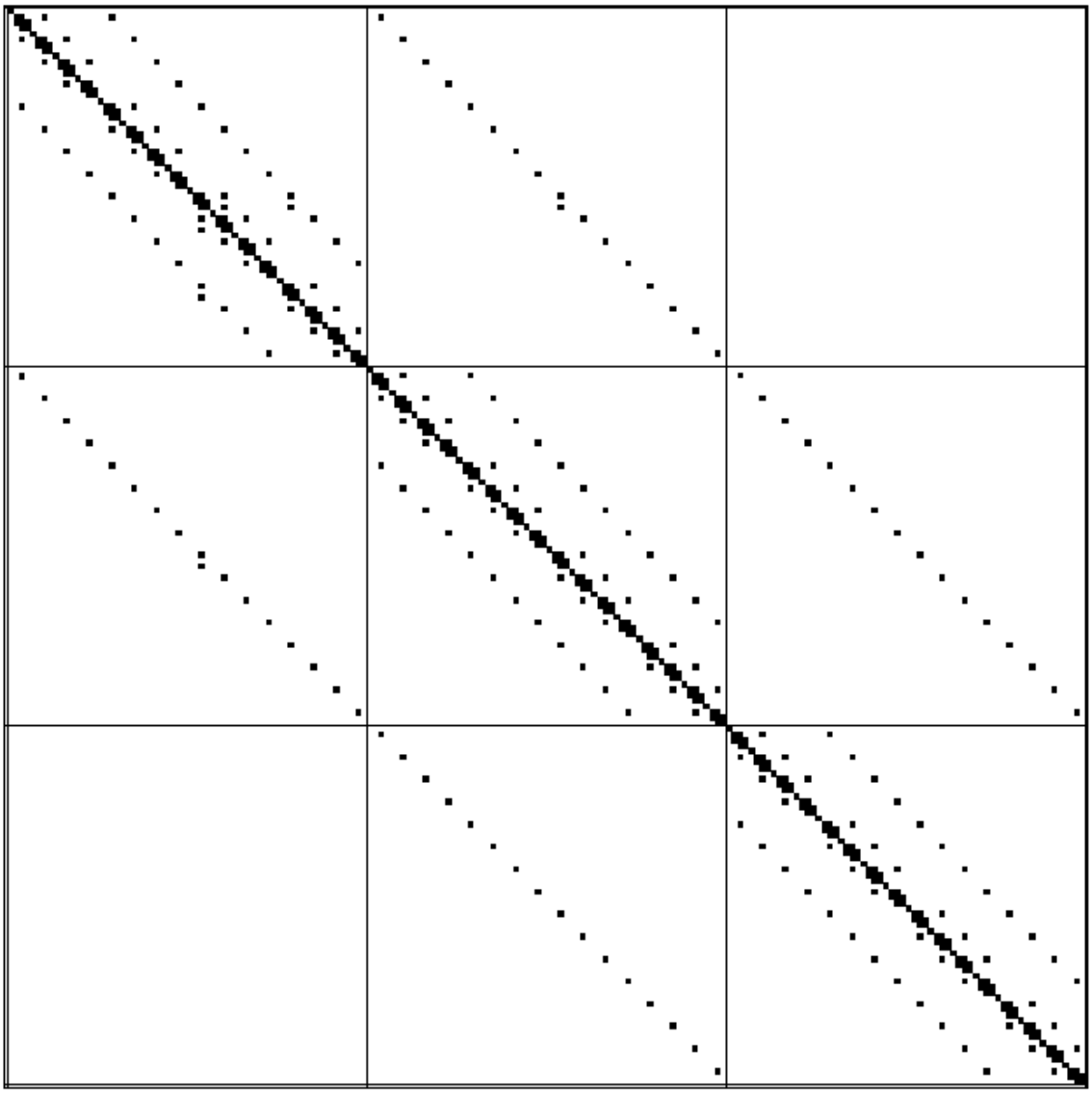

Figure 5.2.2-1: Heptagonal sub-block sparse pattern for 4x4x3 gridblocks from GPAS

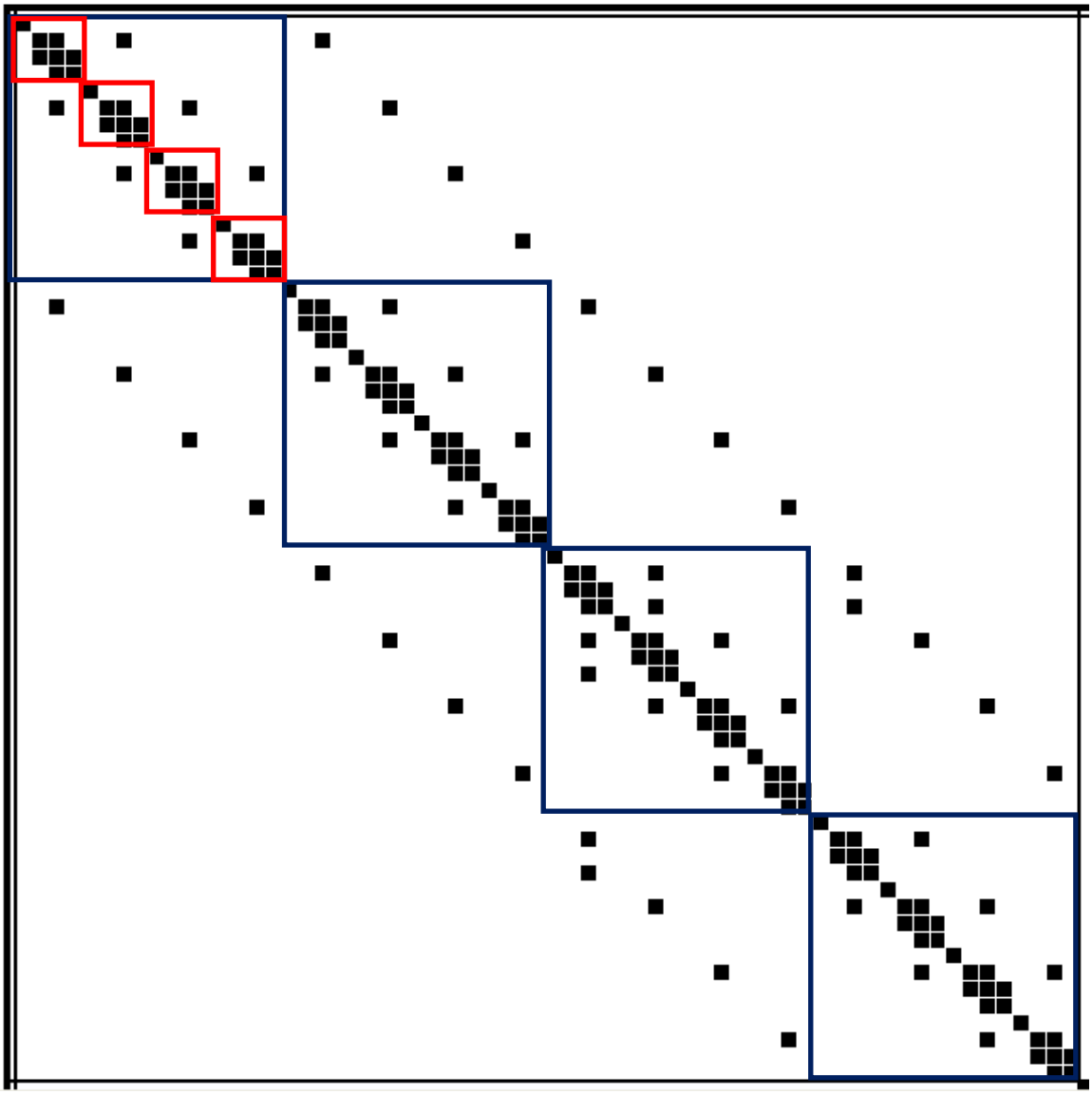

Figure 5.2.2-2: Sparse pattern for one layer with 4x4 gridblocks from Figure 5.2.2-1.

5.3 Current Solvers in GPAS

GPAS employs the PETSc (The Portable, Extensible Toolkit for Scientific computation) package that provides parallel and serial linear solvers, which was developed and maintained by the mathematics and computer science division at Argonne National Laboratory (http://www.mcs.anl.gov/petsc/petsc-as). We used the default choice of PETSc's sub-block matrix solver to solve GPAS linear system equations.

5.4 Implementation of SAMG in GPAS

There are two sub-sections in this section. We will point out the location of SAMG in GPAS' flowchart, and we will describe the details of SAMG's implementation in GPAS.

5.4.1 Location of SAMG in GPAS' Flowchart

The place of SAMG was located in GPAS parallel to that of PETSc in GPAS (Figure 5.4.1-1).

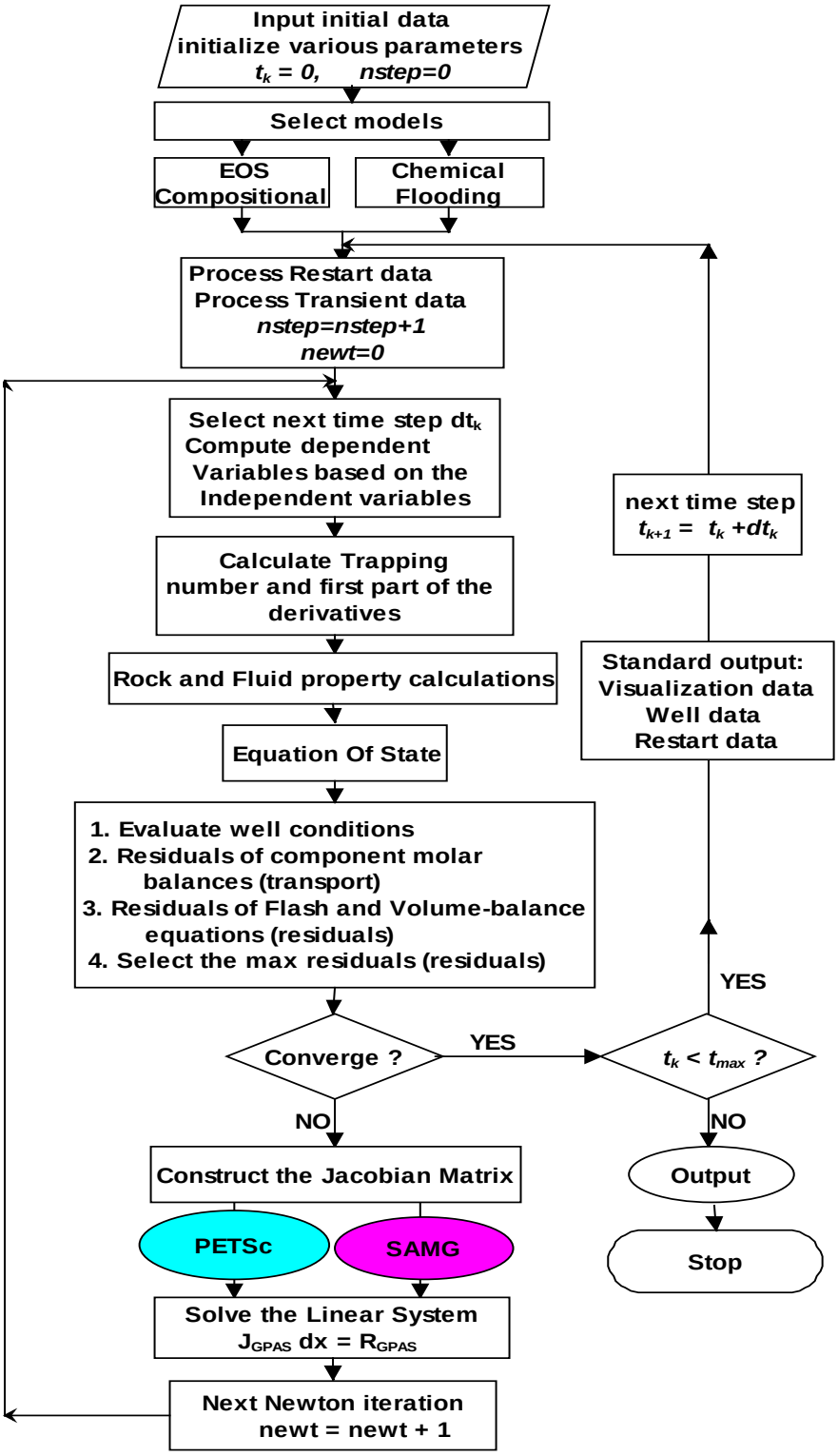

Figure 5.4.1-1: Location of SAMG in GPAS.

5.4.2 Implementation of SAMG in GPAS

There are four steps to add SAMG into GPASv3.5 for both the EOS compositional module and the chemical flooding module.

Step one: Add the index of selecting either PETSc or SAMG. In the GPAS input file, the options of linear solvers was created as follows:

```
LSOLVER:0 petsc,1: samg
LSOLVER = 1
```

Figure 5.4.2-1: Linear solver's options for GPAS.

After adding the linear solver' option into the input file, the source may be modified to read and set the index for the solver's options, which is presented in Figure 5.4.2-2 and Figure 5.4.1-3.

```
COMMON /CONTROL/TIM,TIMOLD,DELTIM,RTIMIN,ACTTIM,DTIMOUT,DTIMRES,
   & DVISOUT,
   & DTIMMUL,DTIMMAX,DTIMMIN,DTIMTOL,NACTTIM,MODEL,NUMPRC,MYPRC,
   & MYPID,NSTEP,NFIN,NFOUT,NFRIN,NFROUT,NFUTIL,NFBUG,LEVELA,LEVELB,
   & LEVELC,LEVELD,LEVELE,BUGOPEN,SPLNOUT,GEOMOUT,LEVERR,BUGKEY,
   & MSGTAG,DATTL,DUALFLAG,
   + LSOLVER,
   & ….
```

Figure 5.4.2-2: Linear solver's index in "control.h" for GPAS.

The functionality of reading "LSOLVER" from an input file was implemented into "idata.F" file represented in Figure 5.4.2-3.

```
LSOLVER = 99
CALL GETVAL('LSOLVER ',LSOLVER,'I4',0,0,0,0,NDUM,NERR)
```

Figure 5.4.2-3: Reading the value of “LSOLVER” in “idata.F” for GPAS.

The solver option selected was indicated as an output information on the screen. The implementation of this part was conducted in “fluid0.F.”

```
if(LSOLVER .eq. 1) then
   write(*,*)'--SAMG    Linear solver--','LSOLVER =', LSOLVER
else
   write(*,*)'--PETSc Linear solver--','LSOLVER =', LSOLVER
endif
```

Figure 5.4.2-4: Output the selection of “LSOLVER” in “fluid0.F” for GPAS.

Step two: Connecting SAMG with GPAS for both the chemical flooding and EOS compositional modules was carried out in “xsolver.F” shown in Figure 5.4.1-4 and Figure 5.4.2-5.

```
if( LSOLVER.eq.1) then
  CALL JSAMG3D(NC,NP,NPH,P,REL,AWW6,AMK6,AMNP6,
   +       AFK,AFW,AFNP,AVK,AVW,AVNP,ARF,ARV,AMK,AMNP,AWW,ARM,ARW,
   +           NX,NY,NZ,I1,I2,J1,J2,K1,K2,KEYOUT,IDX,AMW,AWKN,
   +           AMW6,AWKN6,NA)
else
   CALL JPETSC3D(NC,NP,NPH,P,REL,AWW6,AMK6,AMNP6,
   +           AFK,AFW,AFNP,AVK,AVW,AVNP,ARF,ARV,AMK,AMNP,AWW,ARM,ARW,
   +           NX,NY,NZ,I1,I2,J1,J2,K1,K2,KEYOUT,IDX,AMW,AWKN,
   +           AMW6,AWKN6,NA)
endif
```

Figure 5.4.2-4: Connecting SAMG solver in “xsolver.F” for the EOS compositional module in GPAS.

```
      if( LSOLVER.eq.1) then
            CALL JSAMG3DT(NC,NP,NPH,P,REL,AWW6,AMK6,AMNP6,
     +        AFK,AFW,AFNP,AVK,AVW,AVNP,ARF,ARV,AMK,AMNP,AWW,ARM,ARW,
     +        NX,NY,NZ,I1,I2,J1,J2,K1,K2,KEYOUT,IDX,AMW,AWKN,
     +        AMW6,AWKN6,NA,PVMOL)
      else
            CALL JPETSC3DT(NC,NP,NPH,P,REL,AWW6,AMK6,AMNP6,
     +        AFK,AFW,AFNP,AVK,AVW,AVNP,ARF,ARV,AMK,AMNP,AWW,ARM,ARW,
     +        NX,NY,NZ,I1,I2,J1,J2,K1,K2,KEYOUT,IDX,AMW,AWKN,
     +        AMW6,AWKN6,NA,PVMOL)
      endif
,
       +    AFK,AFW,AFNP,AVK,AVW,AVNP,ARF,ARV,AMK,AMNP,AWW,ARM,ARW,
       +        NX,NY,NZ,I1,I2,J1,J2,K1,K2,KEYOUT,IDX,AMW,AWKN,
```

Figure 5.4.2-5: Connecting SAMG solver in "xsolver.F" for the chemical flooding.

Step three: The data structure of the Jacobian matrix solver was transferred into SAMG's data structure in newly created subroutines for both JSAMG3D for EOS compositional module and for JSAMG3DT for the chemical flooding module, which were located in the "samggpas.F" file. The key difference for data structures between the EOS compositional module and the chemical flooding module is that the chemical flooding module has the unknowns of aqueous components in addition to the same unknowns of hydrocarbon components for GPASv3.5, so the number of unknowns is $2(n_c + n_a) + 2$ instead of $2n_c + 2$.

Step four: The preparation of the parameters of SAMG and calling SAMG were conducted in a newly created subroutine SAMG_sol in the "samgsolver.F" file.

We can conclude the following from the implementation of SAMG into GPAS:

1) Implementation of SAMG is a challenge. It requires both understanding of different modules of GPAS and SAMG's solver procedure.
2) Implementation of SAMG into GPAS has the following advantages:

a. Compatible with the origin code so that it can run with an old data files without any modification.
b. Extensible to new feature of the code so that it can run with a new input data files.
c. Little modification is of the required GPAS code for the implementation of SAMG.

5.5 Numerical Results

In this section, we present the case studies to verify the efficiency of SAMG against PETSc in GPAS using four problems.

5.5.1 Gas Injection with a One-Component Fluid Reservoir (Case I)

A test of gas injection for one component reservoir will be presented in this section. The reservoir properties and component properties and simulation parameters are shown in Table 5.5.1-1.

Table 5.5.1-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Number of gridblocks (NX*NY*NZ)	100×100×3
Grid block size	80×80× (20;30;50) ft^3
Number of hydrocarbon components	1
Component names	C_1
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	730 days
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.17
Initial reservoir pressure	1500 psi
Injection rate	500 mscf/day
Injection contents	100% C_1 (gas)
Bottomhole production pressure	1300 psi
Porosity	0.35
Permeability	10 md

Figure 5.5.1-1 shows the reservoir region for Case I and injection/production wells. There are two wells, one injection well and one production well. The injection well injected gas

C_1 with a constant injection rate of 500 mscf/day. The production well was kept at a constant bottomhole pressure of 1,300 psi.

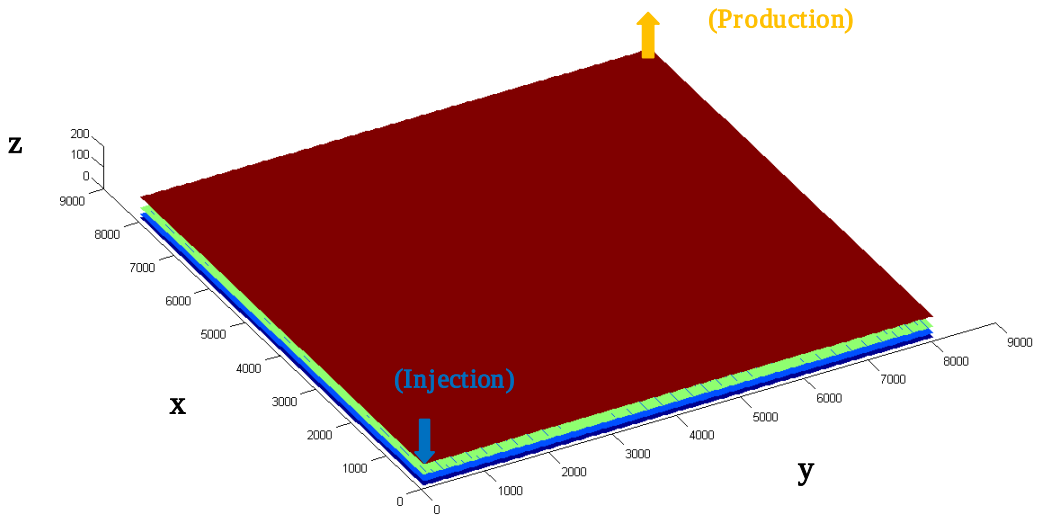

Figure 5.5.1-1: Reservoir model for Case I.

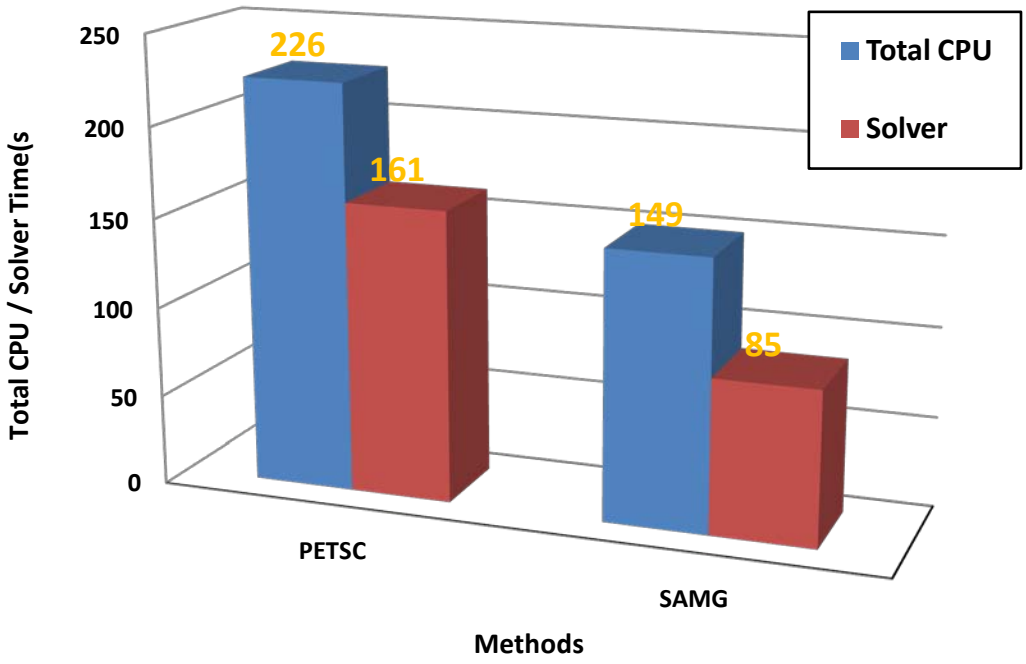

Figure 5.5.1-2: Comparison of running times of Case I.

Figure 5.5.1-2 shows the different CPU times for Case I using GPAS for two linear solvers, PETSc and SAMG. The results show that SAMG is faster than PETSc solver for this case, i.e., SAMG took less than 52% of the running time of PETSc.

5.5.2 Water Injection with a One-Component Fluid Reservoir (Case II)

A test of water injection for a one-component reservoir will be presented here. Reservoir properties and simulation parameters are shown in Table 5.5.1-2.

Table 5.5.2-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Number of gridblocks (NX*NY*NZ)	100×100×3
Grid block size	80×80× (20;30;50) ft^3
Number of hydrocarbon components	1
Component names	C_{10}
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	3650 days
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.17
Initial reservoir pressure	1500 psi
Injection rate	500 mscf/day
Injection contents	100% water
Bottomhole production pressure	1300 psi
Porosity	0.35
Permeability	10 md

The reservoir region of Case II is the same as that of Case I. There are two wells, one injection well and one production well. The injection well injected water with a constant injection rate of 500 mscf/day. The production well was kept at a constant bottomhole pressure of 1,300 psi.

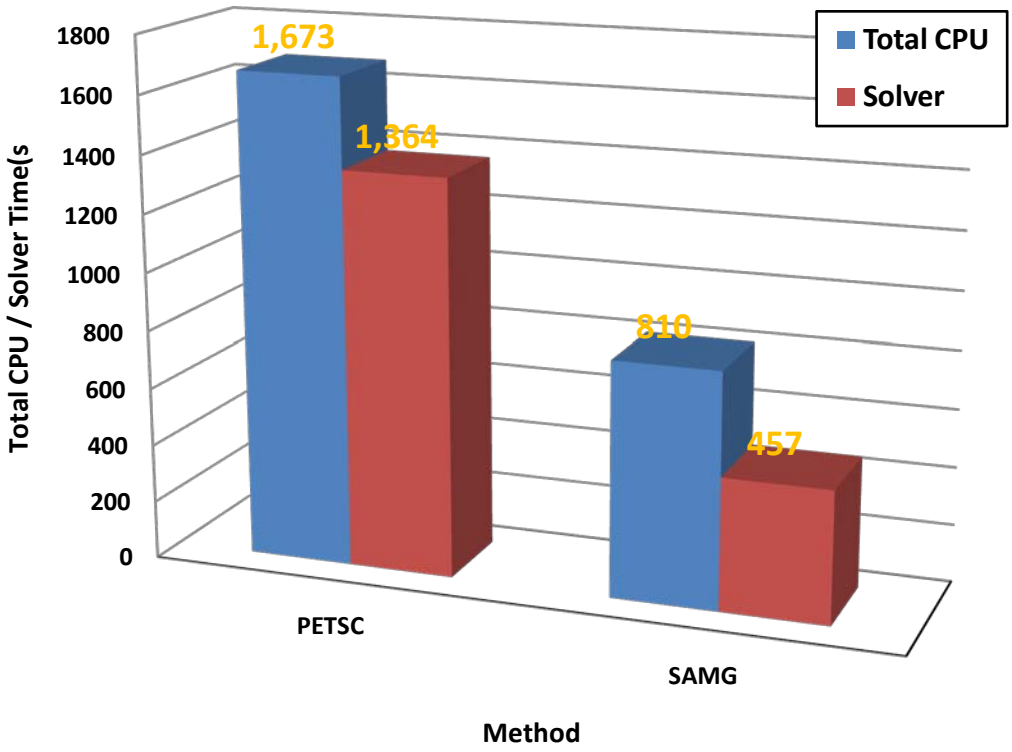

Figure 5.5.2-1: Comparison of running times for Case II.

Figure 5.5.2-1 shows CPU times for Case II using GPAS with two different solvers, PETSc and SAMG. The CPU times in Figure 5.5.2-1 show that SAMG is the fastest solver, which took less than 35% of the running time of PETSc.

5.5.3 Water Injection with a Two-Component Fluid Reservoir (Case III)

A test of water injection for a two-component reservoir will be presented here. Reservoir and simulation parameters are shown in Table 5.5.3-1.

Table 5.5.3-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Number of gridblocks (NX*NY*NZ)	100×100×3
Grid block size	80×80× (20;30;50) ft^3
Number of hydrocarbon components	2
Component names	C_{10} ,C_{15}
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	3650 days
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.17
Initial reservoir pressure	1500 psi
Injection rate	700 mscf/day
Injection contents	100% water
Bottomhole production pressure	500 psi
Porosity	0.35
Permeability	10 md

The reservoir region of Case III is the same as that of Case I. There are two wells, one injection well and one production well. The injection well injected water with a constant injection rate of 700 mscf/day. The production well was kept at a constant bottomhole pressure of 500 psi.

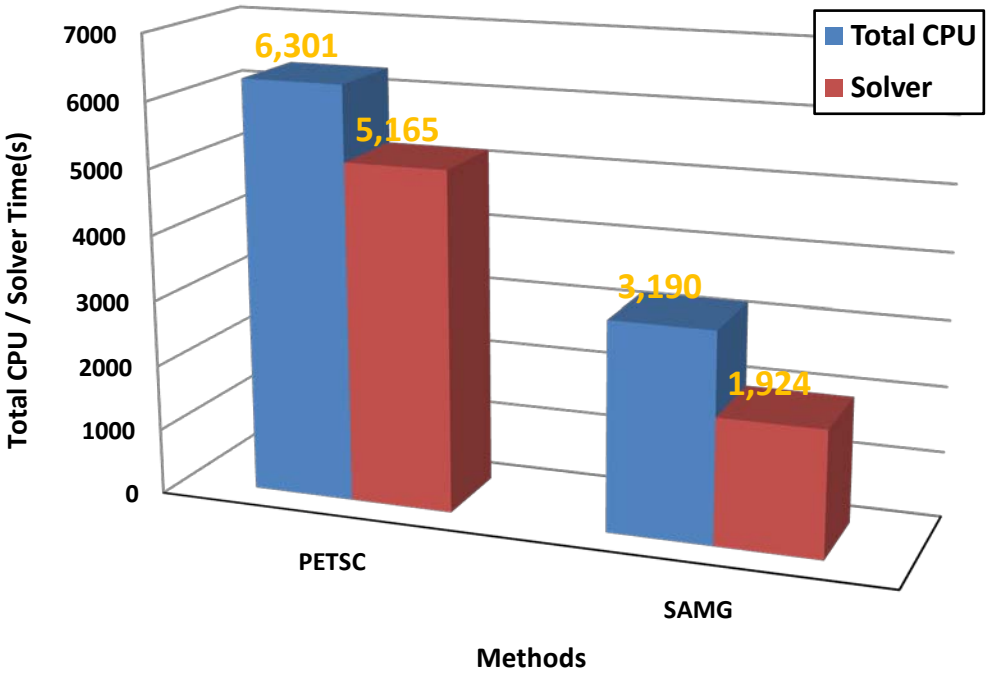

Figure 5.5.3-1: Comparison of running times for Case III.

Figure 5.5.3-1 shows the CPU times for Case III using GPAS with two solvers, PETSc and SAMG. The CPU times in Figure 5.5.3-1 shows that SAMG is the faster solver, which took less than 38% of the running time of PETSc.

5.5.4 Water Injection with a Three-Component Fluid Reservoir (Case IV)

A test of water injection for a three-component reservoir will be presented next. The reservoir and simulation parameters are shown in Table 5.5.4-1.

Table 5.5.4-1: Reservoir properties and simulation parameters.

Parameter	**Value**
Number of gridblocks (NX*NY*NZ)	100×100×3
Grid block size	80×80× (20;30;50) ft^3
Number of hydrocarbon components	3
Component names	C_{10} ,C_{15}, C_{20}
Maximum number of phases	3
Number of wells	2
Type of wells	Injection well, production well
Simulation time	3650 days
Water viscosity	1 cp
Water density	62.46 lb/ft^3
Initial water saturation	0.17
Initial reservoir pressure	1,500 psi
Injection rate	700 mscf/day
Injection contents	100% water
Bottomhole production pressure	500 psi
Porosity	0.35
Permeability	10 md

The reservoir region is the same as that of Case I. There are two wells, one injection well and one production well. The injection well injected water with a constant injection rate of 700 mscf/day. The production well was kept at a constant bottomhole pressure of 500 psi.

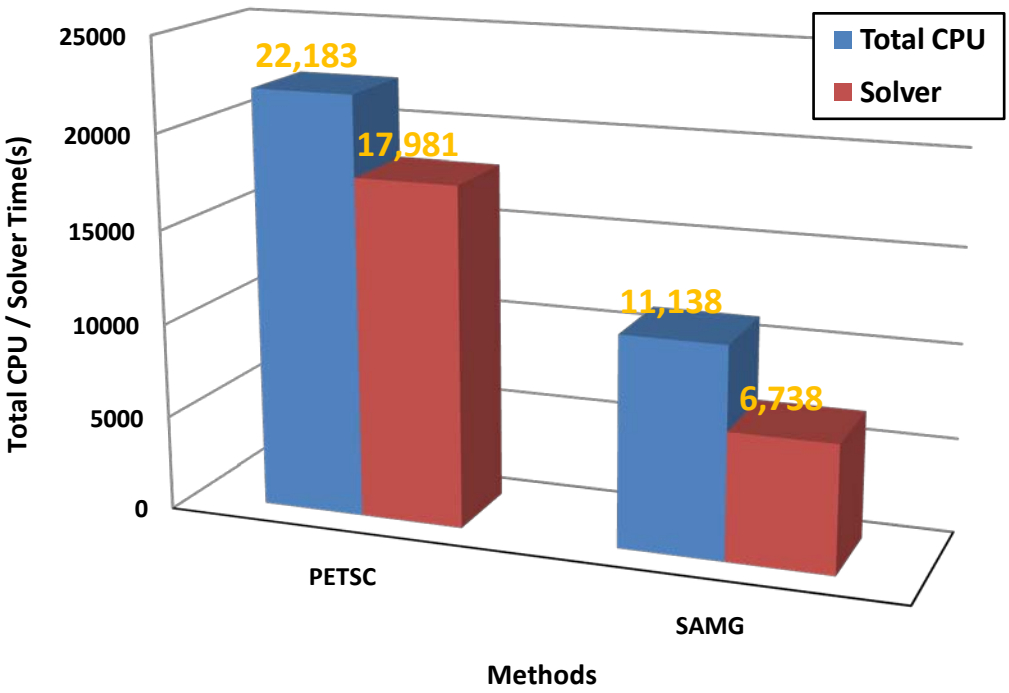

Figure 5.5.4-1: Comparison of running times of Case IV

Figure 5.5.4-1 shows the CPU times for Case IV using GPAS with two solvers, PETSc and SAMG. The CPU times on Figure 5.5.2-1 show that SAMG is the faster solver, which took less than 39% of the running time of PETSc.

5.5.5 Comments and Conclusions

Several conclusions may be drawn from the numerical results from Section 5.5.1 through Section 5.5.4: SAMG took about 52% and 35% of PETSc's computational time for one component in both the gas and water flooding cases, respectively, which may mean that SAMG can solve the water flooding case better than the gas flooding case. SAMG took about 38% and 39% of PETSc's CPU time for the two-component and three-component water flooding tests, which means that SAMG kept the same advantage against PETSc in GPAS for the cases with the increased number of reservoir fluid components.

SAMG treats sub-block matrices differently. Based on SAMG grid-based method, SAMG has special improvement on smoother and interpolation operators. However, PETSc doesn't have grid-based concept, so PETSc will view all kinds of matrices as a scalar matrix. Thus, the solving speed may not be efficient all cases, especially for the sub-block matrices. This may be the reason why SAMG is faster than PETSc in the four cases.

.

Chapter 6: A Partition Method with a Thermal Module for GPAS

In this chapter, a new implementation of a thermal module using a partition method (PM) for GPAS will be presented in order to investigate the efficiency and flexibility of the partition method.

6.1 Introduction

6.1.1 Literature Review for Thermal Simulation

The thermal simulations based on the Equation-of-State (EOS) have been studied for more than two decades. Ishimoto was the first to develop a one-dimensional and fully implicit compositional steam-flood simulator (Ishimoto 1985, Ishimoto et al. 1987). A fully implicit, four phase multi-component, multidimensional steam and combustion simulator, in which a fully implicit well model and appropriate and robust iterative techniques for solving larger thermal problems were included, was presented by Rubin and Buchanan in 1985. Chein et al. (1989) developed a general purpose compositional simulator in which the thermal model selects K-values or the EOS to calculate fluid properties. Trangenstein analyzed a two-component, three-phase flow thermal model of the effect of thermodynamic principles on flow equations in 1989. Considering water as a non-ideal component and enthalpy as a primary variable, Brantferger et al. 1991 developed a novel simulator.

6.1.2 Approaches for the Implementation of a Thermal Simulator

There are different approaches for the implementation of EOS-based thermal compositional simulators. A fully coupled, fully implicit oil reservoir simulator, in which its frame-work supports an IMPES and a sequential semi-implicit formulation, was developed by Miffin et al. in 1991. Chan and Sarioglu described a procedure for incorporating fracture characteristics in a thermal reservoir simulator in 1992. Cicek developed a compositional, fully-implicit 3D simulator that can handle steam injection.

In 2005, Cicek also improved his simulator so that it can handle naturally fractured reservoirs. He tested the steam displacement of oil in a naturally fractured reservoir and investigated a two-phase liquid region by a steam condensation front. Varavei and Sepehrnoori (2009) described the development of a fully implicit EOS thermal flooding simulator. In their simulator, the thermodynamic properties and compositional effects for the phase behavior calculation are more accurate than those produced by previous simulators because the physical properties for phase and components are calculated based on EOS, not on any steam-tables.

However, conventional fully implicit EOS modules are highly integrated with other modules in the simulators which prohibited flexibility, and maintaining a high level of efficiency. This chapter presents a new implementation of a thermal module into the fully implicit isothermal GPAS. The new implementation differs from the traditional fully coupled method since the formulation can allow the flexibility of adding or removing the thermal module with the use of existing simulators. In addition, the new implementation enhances the calculation efficiency so that the new implementation is faster than the traditional fully implicit method.

6.2 Mathematical Model Equations of GPAS and a Thermal Module

6.2.1 Mathematical Model Equations of Isothermal GPAS

GPAS version 3.5, using a traditional fully implicit method with a finite difference discretization on structure grid system, has been developed by the reservoir simulation group of CPGE (The Center for Petroleum and Geosystems Engineering) at The University of Texas at Austin (Wang et al. 1999, Han et al. 2005, Wang et al. 1997). The isothermal compositional model of GPAS can be described by the following mathematical models in the PDEs forms for n_c hydrocarbon components and n_p hydrocarbon components without considering mass fluxes between the water phase and other phases.

Mass conservation equations for hydrocarbon component is given by

$$V_b \frac{\partial(\phi N_i)}{\partial t} + V_b \nabla \cdot \sum_{j=a,v,o} (x_{ij}.\xi_j.\vec{v}_j) + R_i = \dot{q}_i \qquad i = 1... \; n_c + 1. \qquad (6.2.1\text{-}1)$$

The term $\dot{q}_i$ is a molar rate for either source or sink of the given component and R_i is the reaction rate term of component i. The subscripts of a, v, and o represent aqueous, vapor and oleic phases, respectively. N_i is the number of moles of i component per pore volume, x_{ij} is the mole fraction of component i in phases j, ξ_j is the molar density of phase j, and $\vec{v}_j$ is the volumetric flux of phase j, which follows the Darcy's law as follows:

$$\vec{v}_j = -\vec{K}\lambda_{rj}\left(\nabla P_j - \gamma_j \nabla D\right), \qquad (6.2.1\text{-}2)$$

where $\vec{K}$ is the permeability tensor defined as

$$\vec{K} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix}, \qquad (6.2.1\text{-}3)$$

λ_{rj} is phase relative mobility defined as $\lambda_{rj} = \frac{k_{rj}}{\mu_j}$, and γ_j is specific weight of phase *j*. The Peng-Robinson EOS is used for phase behavior calculation.

Phase equilibrium equations of a three-phase model consisting of two fugacity equations are given by

$$\begin{aligned} f_i^o - f_i^g &= Ln\varphi_{i,o} - LnK_{1,i} - Ln\varphi_{i,g} = 0 \qquad i = 1,2,...n_c + 1 \\ f_i^o - f_i^a &= Ln\varphi_{i,o} - LnK_{2,i} - Ln\varphi_{i,a} = 0 \qquad i = 1,2,...n_c + 1 \end{aligned}, \qquad (6.2.1\text{-}4)$$

where superscripts a, v, and g represent aqueous, oleic and gaseous phases, respectively.

The pore volume constraint Equation is given as

$$\sum_{j=a,v,o} \left[\frac{N_j}{\zeta_j}\right] = 1.0 \; . \qquad (6.2.1\text{-}5)$$

6.2.2 A Thermal Module for a Partition Method

An energy conservation equation is given by

$$V_b \frac{\partial U^T}{\partial t} + V_b \nabla \cdot \sum_{j=a,v,o} \zeta_j h_j \vec{v}_j - V_b \nabla \cdot \left(\lambda_T \nabla T \right) = -\dot{Q}_L + \dot{q}_H \quad , \tag{6.2.2-1}$$

where $U^T = (1-\phi)\zeta_r U_r + \phi \sum_{j=a,v,o} \zeta_j S_j u_j$ is the sum of rock and total fluid internal energy per bulk volume; h_j is the phase molar enthalpy, λ_T is the effective conductivity coefficient, ζ_j and S_j are the phase molar density and phase saturation, ζ_r is the rock density, ϕ is the porosity, *T* represents the temperature, $\dot{Q}_L$ is the heat loss rate, and $\dot{q}_H$ is the rate of heat injection.

6.3 Partition Method

6.3.1 Introduction of a Partition Method

Prevost et al. (1997) proposed an implicit unconditionally stable partitioned solution procedure for the simultaneous integration of transient coupled field problems. We can apply this procedure for our more complicated implementation of EOS for our GPAS simulator. This procedure was used in GPAS to couple a thermal module with the isothermal GAPS version code. The implementation does not require the fully coupled system equations to be merged together. We employed a Krylov Subspace iterative procedure to equip the new implementation to avoid forming and assembling the Schur complement matrix in order to reduce the memory and computational requirements.

6.3.2 Comparison with the Classical Fully Implicit Method

For the classical fully implicit method (CFIM), the governing equations (6.2.1-1-6.2.1-5) are discretized by the FDM. There are $3n_c + 2$ non-linear equations and unknowns to be solved at each gridblock in the case where three phases are in equilibrium. The unknowns or the primary variables are as follows:

$$P_w, N_w, N_1, N_2, ..., N_{nc}; LnK_{1,og}, LnK_{2,og}, ..., LnK_{nc,og}, LnK_{1,ol_2}, LnK_{2,ol_2}, ..., LnK_{nc,ol_2}, T$$

By applying the Newton method to these nonlinear equations, a traditional coupled Jacobian linear system will be obtained from each Newton's iteration as follows:

$$\begin{pmatrix} J_{1,1} & J_{1,2} & \cdots & J_{1,n_b} \\ J_{2,1} & J_{2,2} & \cdots & J_{2,n_b} \\ \vdots & & \vdots & \\ J_{n_b,1} & J_{n_b,2} & \cdots & J_{n_b,n_b} \end{pmatrix} \cdot \begin{pmatrix} \Delta\vec{x}_1 \\ \vdots \\ \Delta\vec{x}_2 \\ \vdots \\ \Delta\vec{x}_{n_b} \end{pmatrix} = - \begin{pmatrix} \vec{R}_1 \\ \vdots \\ \vec{R}_2 \\ \vdots \\ \vec{R}_{n_b} \end{pmatrix}, \qquad (6.3.1\text{-}1)$$

where $J_{i,j}$ is "tight" coupled as shown in Figure 6.3.2-1.

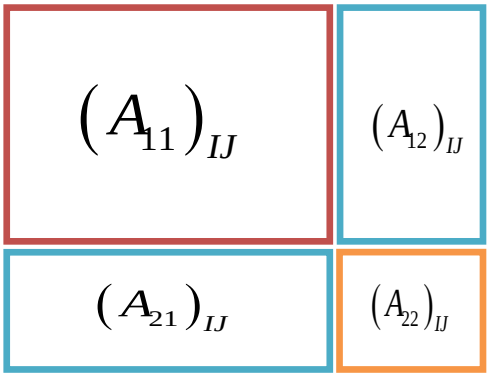

Figure 6.3.2-1: The tight structure of sub-matrix at (I, J) for the fully implicit method.

The detailed structure of the Jacobian matrix is given in Figure 6.3.2-2.

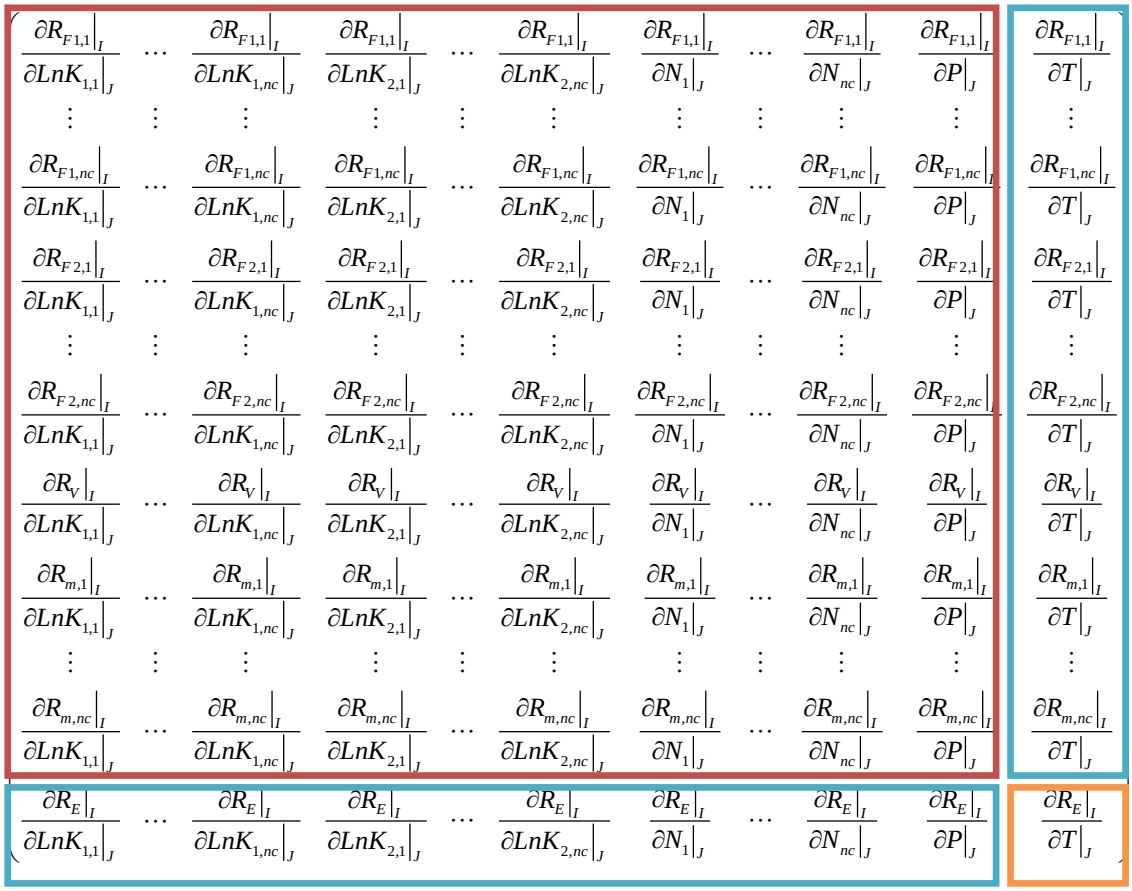

Figure 6.3.2-2: Tight structure of sub-block matrix at (I, J) for the fully implicit method.

The partition method can use flexible coupling structures, so A_{11} can be separated from A_{22}, A_{12}, and A_{21} in the program structure, in which GPAS calculates A_{11} only explicitly, but A_{22}, A_{12}, and A_{21} have fewer requirements than those of the traditional method so that they do not need to be explicitly calculated, but just stored in the new module in another independent part of code with the basic functionality such as a matrix-vector multiplication. The basic functionality of A_{22}, A_{12}, and A_{21} is that, at each time step, y_1, y_2, and y_3 can be calculated from $y_1 = A_{21}z_1$, $y_2 = A_{12}z_1$, and $y_3 = A_{22}z_1$

when the vectors z_1 and z_2 are given. Here, A_{11} represents GPAS software, which needs to be modified for extra functionality so that it can solve x from $A_{11}x = b$ when b is available at a given time step. Then the partition method can be used.

In the classical approach, the new coupling system from a Newton's iteration is given in the following:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \delta_{GPAS} \\ \delta_{NewM} \end{pmatrix} = \begin{pmatrix} b_{GPAS} \\ b_{NewM} \end{pmatrix}, \qquad (6.3.2\text{-}2)$$

where A_{11} and b_{GPAS} are matrices and vectors from original GPAS modules: A_{22} and b_{NewM} are matrices and vectors from the thermal modules; A_{12} and A_{21} are coupled matrices between the original GPAS modules and the new thermal module.

The flowchart for GPAS and the thermal module is given in Figure 6.3.2-3. In this approach, the thermal module is fully coupled and solved with other PDEs in GPAS, where $A_{11} = J_{GPAS}$, $A_{22} = J_T$, $A_{12} = J_{GPAS\_T}$, $A_{21} = J_{T\_GPAS}$, $b_1 = -R_{GPAS}$, and $b_2 = -R_T$. The implementation will modify almost every part of GPAS code so that it may more likely to cause errors in the existing GPAS simulator, and also, once it is done, it is not easy to remove it from the implementation.

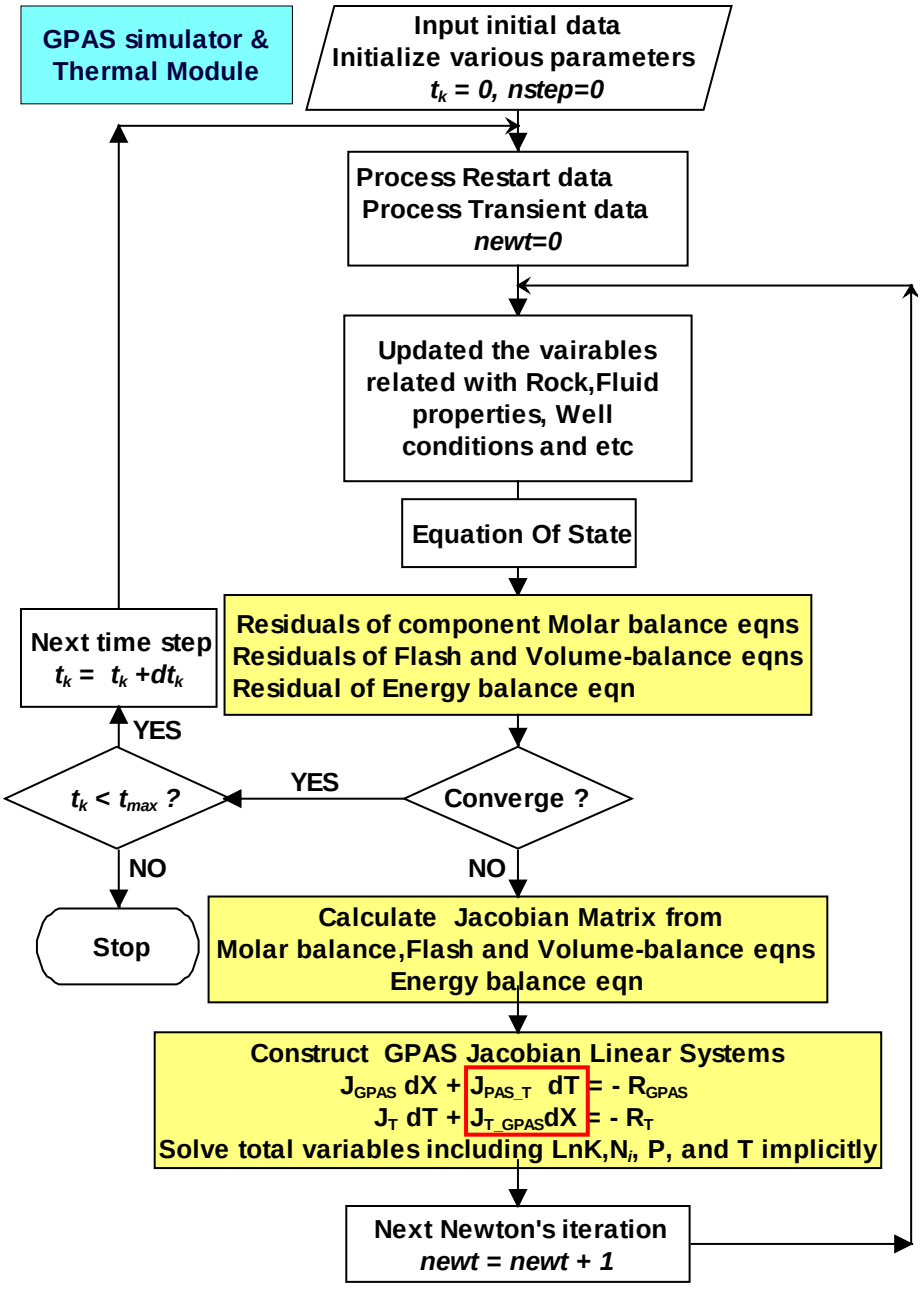

Figure 6.3.2-3: Adding a thermal module into GPAS using fully implicit method.

The fully coupled new Jacobian linear systems for original equations in GPAS and thermal module will be formed as

$$\begin{pmatrix} J_{GPAS} & J_{GPAS\_T} \\ J_{T\_GPAS} & J_T \end{pmatrix} \begin{pmatrix} dX \\ dT \end{pmatrix} = \begin{pmatrix} -R_{GPAS} \\ -R_T \end{pmatrix}. \quad (6.3.2\text{-}3)$$

6.3.3 Implementation of a Partition Method

The partition method with Schur complement employs a different approach from the classical method. The thermal module can be built independently of the GPAS simulator while keeping the same structure of GPAS. In order to couple thermal module with GPAS, the first step is making the connection at the "update the variables," "converge," and "construct Jacobian matrix" phases between GPAS and the thermal module to keep them at the same Newtonian iteration step and the same time step. The second step is coupling the Jacobian matrices with the same effect of the fully implicit method, but solving them separately in two different modules such as GPAS and thermal module as shown in Figure 6.3.3-1.

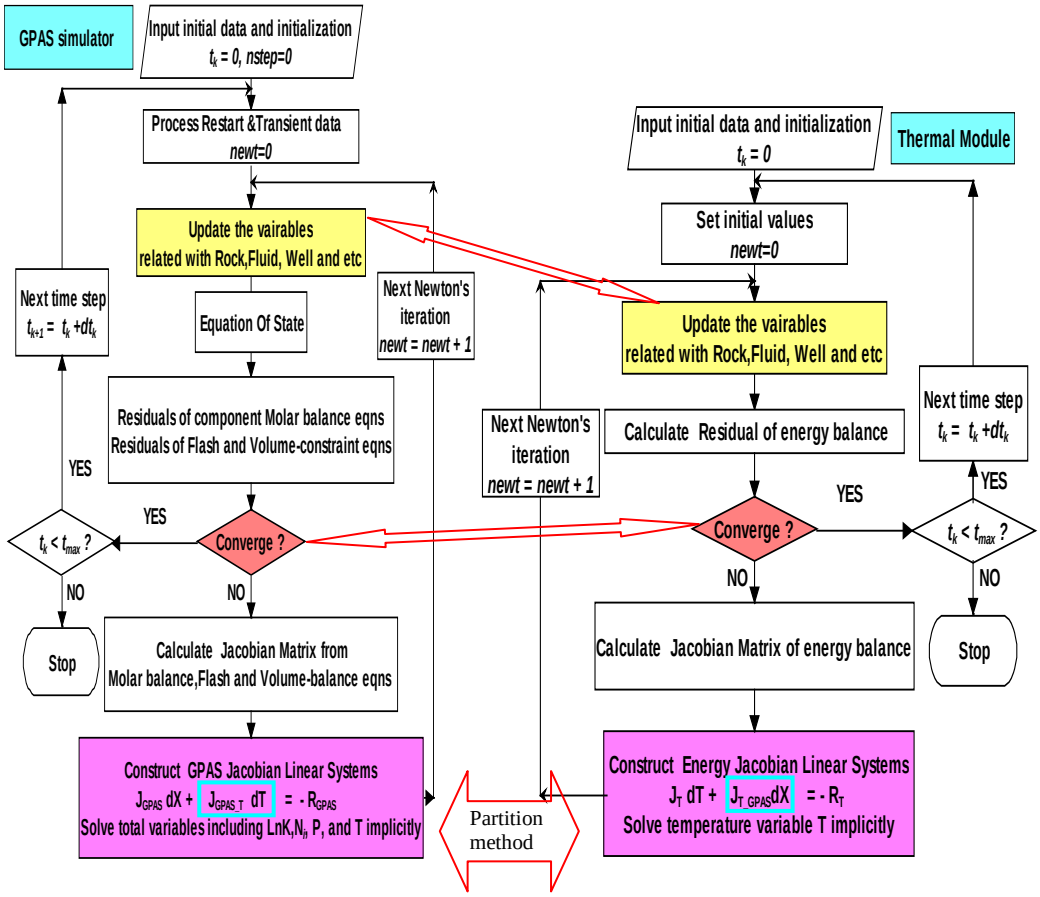

Figure 6.3.3-1: Adding a thermal module through a partition method.

Instead of solving all of the linear systems, we can transform them into the following equivalent linear systems by the block Gauss elimination:

$$\begin{pmatrix} A_{11} & A_{12} \\ o & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix} \begin{pmatrix} \delta_{GPAS} \\ \delta_{NewM} \end{pmatrix} = \begin{pmatrix} b_{GPAS} \\ b_T - A_{21}A_{11}^{-1}b_{NewM} \end{pmatrix}, \qquad (6.3.3\text{-}1)$$

where $A_{22} - A_{21}A_{11}^{-1}A_{12}$ is called the Schur complement. The solution procedure using the partition method can be carried out in two steps:

Step 1 (Backward) solve new module unknowns from the Schur complement system

$$\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)\delta_{NewM} = \left(b_{NewM} - A_{21}A_{11}^{-1}b_{NewM}\right) \qquad (6.3.3\text{-}2)$$

Step 2 (Forward) solve the unknowns of the original GPAS based on the GPAS

$$A_{11}\delta_{GPAS} = \left(b_{GPAS} - A_{12}\delta_{NewM}\right), \tag{6.3.3-3}$$

In actual implementation, we may solve the equations as given in Figure 6.3.3-2.

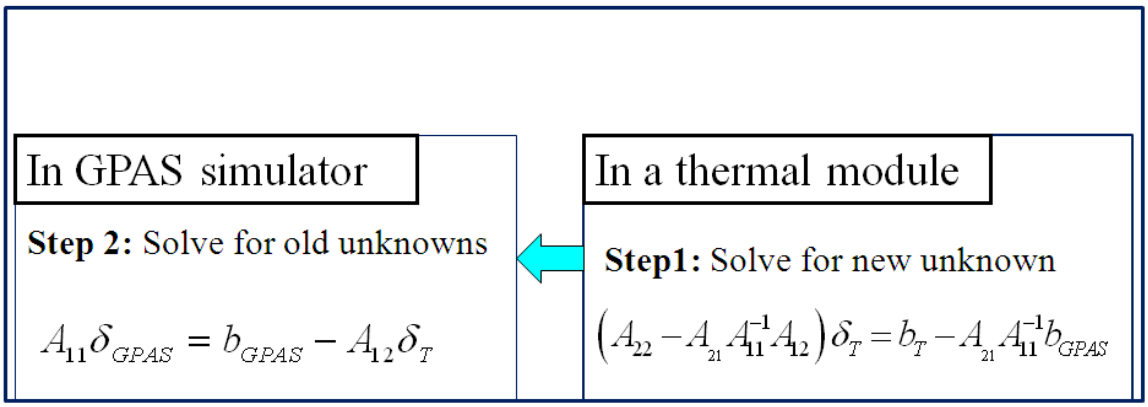

Figure 6.3.3-2: A partition method for adding a thermal module.

Step 2 can be solved once δ_{NewM} is given. However, the challenge is to solve Step 1 because the Schur complement cannot be calculated explicitly due to the high cost of both computation and storage, so the all parts of the Schur complement are kept in a separate sparse form but are combined with those parts to solve the Schur Complement. Considering that A_{11}^{-1} will be needed in the Schur complement system, we will employ a Krylov iterative method, i.e., BiCGSTAB, which only requires matrix-vector multiplication in the solution procedure.

Thus, Step 1 can be performed. The matrix-vector product operation $z = \left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)y$ can be solved as follows:

a) $y_1 = A_{22}y$, $y_2 = A_{12}y$

b) $A_{11}y_3 = y_2$ is solved by GPAS to get $y_3 = A_{11}^{-1}y_2$

c) $y_4 = A_{21}y_3$

d) $z = y_1 - y_4$,

and the entire problem can be solved.

From the above description, it is easy to see that the conventional method has the following disadvantages compared to the partition method:

- The classical method requires that both the source code of GPAS and the thermal module code be available for the implementation.
- The classical method requires that A_{11}, A_{22}, A_{12} and A_{21} be calculated in an explicit form.
- The "tight" coupling structure that a classic approach merges with a new module creates more difficulties to add a new module into GPAS or remove the exiting module from GPAS.
- The newly added module from the conventional approach affects the original software so much that the future debugging procedure will become more complicated.

6.4 Numerical Results

6.4.1 Three Test Cases

In order to validate the partition method, three test cases with different numbers of grid blocks, different components, and different simulation times were simulated in a quarter-of-five pattern.

Case I is a two-component water flooding case consisting of water and C_{17}. The 3D homogeneous reservoir is 240x240x50 ft^3 in size with a permeability of 250 md and a porosity of 0.3. The initial pressure and temperature of the reservoir are 600 psi and 240 ^oF, respectively. The heat capacity ratio for fluid and rock is one. The volumetric heat capacity is 35 Btu/(ft^3-°R) and the thermal conductivity is 35 Btu/(ft-day-°R). The reservoir is divided into 6x6x5 gridblocks. Steam and water with the rate of 1500 Lbmol/day and temperature of 790 °F are injected into the reservoir. The production bottomhole pressure is maintained at 580 psia. The simulation period is 1500 days. The reservoir region of Case I is shown in Figure 6.4.1.-1.

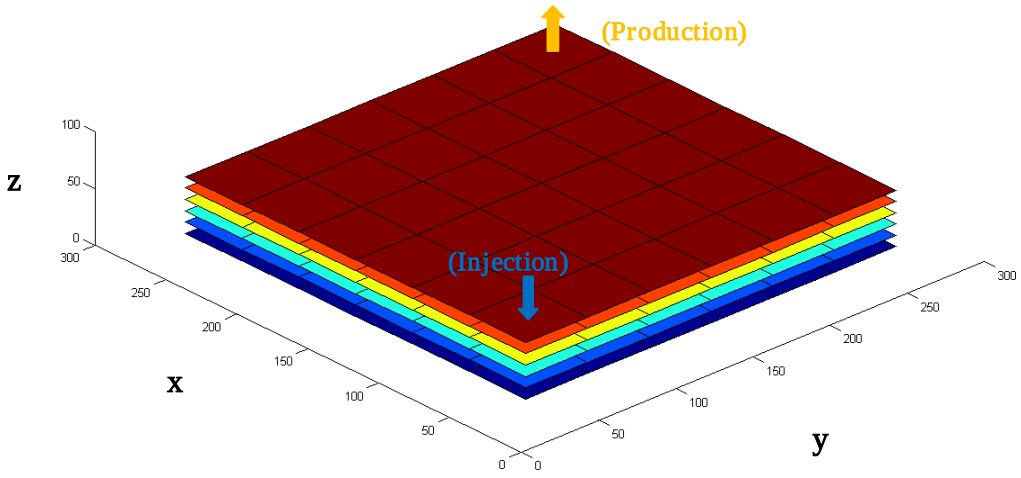

Figure 6.4.1-1: Reservoir model for Case I.

Case II is a four-component case consisting of C_6, C_{17}, $C_{19,}$ and water components. The 3D homogeneous reservoir is 320x320x30 ft^3 in size with a permeability of 150 md and porosity of 0.3. The initial pressure and temperature of the reservoir are 450 psi and 240 ^oF, respectively. The heat capacity ratio for fluid and rock is one. The volumetric heat capacity is 35 Btu/(ft^3-°R) and the thermal conductivity is 35 Btu/(ft-day-°R). The reservoir is divided into 8x8x5 gridblocks. Steam and water at the rate of 2200 Lbmol/day and temperature of 790 °F are injected into the reservoir. The production bottomhole pressure is maintained at 400 psia. The simulation period is 1500 days. The well locations of the reservoir in Case II are same as Figure 6.4.1.-1.

Case III is a five-component case consisting of C_6, C_{10}, C_{15}, C_{17}, C_{19}, and water components. The 3D homogeneous reservoir is 270x270x40 ft^3 in size with a permeability of 150 md and a porosity of 0.3. The initial pressure and temperature of the reservoir are 550 psi and 240 ^oF, respectively. The heat capacity ratio for fluid and rock is one. The volumetric heat capacity is 35 Btu/(ft^3-°R) and the thermal conductivity is 35 Btu/(ft-day-°R). The reservoir is divided into 9x9x4 gridblocks. Steam water with the rate of 2100 Lbmol/day and temperature of 790 °F is injected into the reservoir. The production bottomhole pressure is maintained at 520 psia. The simulation period is 2000 days. The locations of wells in this reservoir are the same as Figure 6.4.1-1.

6.4.2 Numerical Results of the Three Test Cases

Tables 6.4.2-1, 6.4.2-2, and 6.4.2-3 show a comparison of the computational times for the CFIM and a PM for the three cases. CFIM is a new solver sharing the same Krylov subspace method BiCGSTAB and ILUK preconditioner. The partition method also uses the BiCGSTAB and ILUK preconditioner. Table 6.4.2-1 shows that the computational time of CFIM is 24 seconds, which is 5.88% slower than the computational time of PM, which is 23 seconds. Table 6.4.2-2 shows that the computational time of CFIM is 66 seconds, which is 9.24% slower than the computational time of PM, which is 60 seconds. Table 6.4.2-3 shows that the computational time of CFIM is 306 seconds of 9.168% higher than the computational time of PM of 281 seconds. The comparison in Figures 6.4.2-1 and 6.4.2-2 show the trend that with the increasing the number of gridblocks and components, PM is efficient compared to CFIM.

Table 6.4.2-1: Total time and linear solver time for Case I.

	CFIM	PM
Total time (s)	24	23
Solver time (s)	8	5

Table 6.4.2-2: Total time and linear solver time for Case II.

	CFIM	PM
Total time (s)	66	60
Solver time (s)	25	17

Table 6.4.2-3: Total time and linear solver time for Case III.

	CFIM	PM
Total time (s)	306	281
Solver time (s)	135	102

Table 6.4.2-4: Summary of the three cases.

Case	Gridblocks	Components	Injection	Simulation time
1	6x6x5	2	Hot water	1500
2	8x8x5	4	Hot water	1500
3	9x9x4	6	Hot water	2000

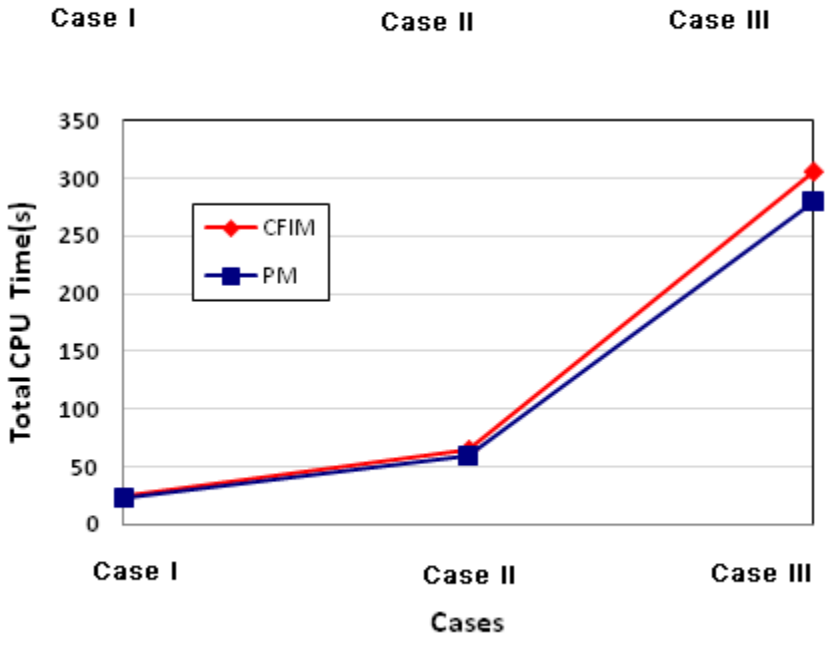

Figure 6.4.2-1: Total CPU time for the three cases.

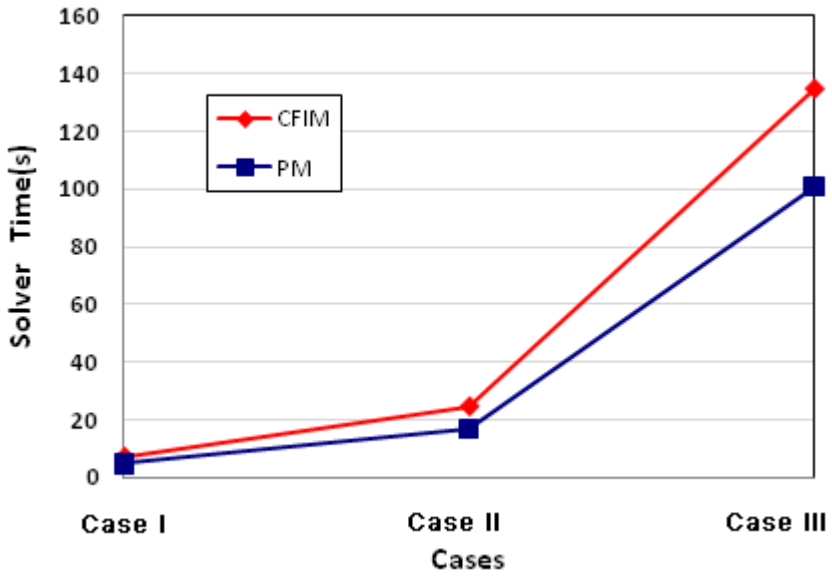

Figure 6.4.1-2: Solver CPU time for the three cases.

The first type of validation curves, i.e., the total oil recovery curves for both CFIM and PM, are shown in Figures 6.4.2-3, 6.4.2-4, and 6.4.2-5. These figures show that PM matches CFIM perfectly. The breakthrough times occurred approximately at the 1197 days, 800 days and 842 days for Cases I through Case III, respectively.

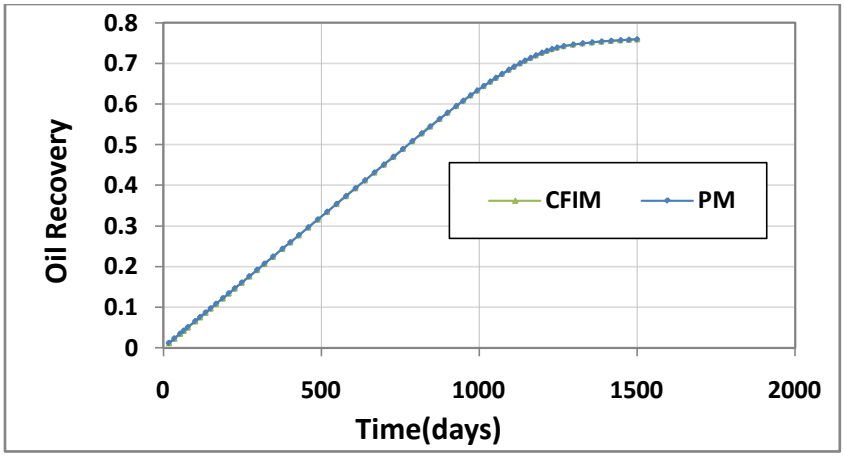

Figure 6.4.2-3: Total oil recovery (STB/day) versus time (day) for Case I.

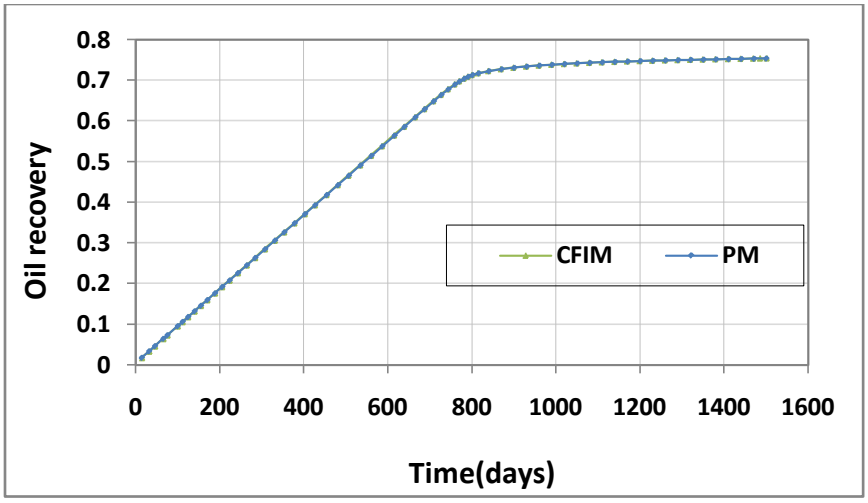

Figure 6.4.2-4: Total oil recovery (STB/day) versus time (days) for Case II.

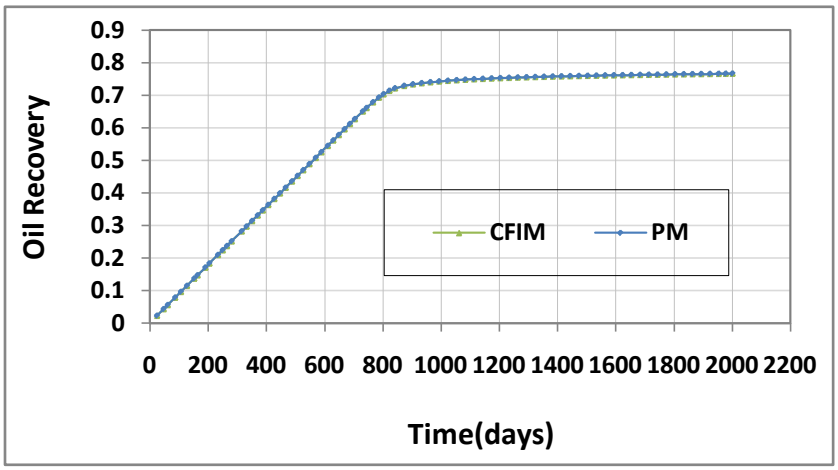

Figure 6.4.2-5: Total oil recovery (STB/day) versus time (days) for Case III.

The second type of validation curves, i.e., the average reservoir pressure curves for both PM and CFIM are shown in Figures 6.4.2-6, 6.4.2-7, and 6.4.2-8. The two curves matched very well.

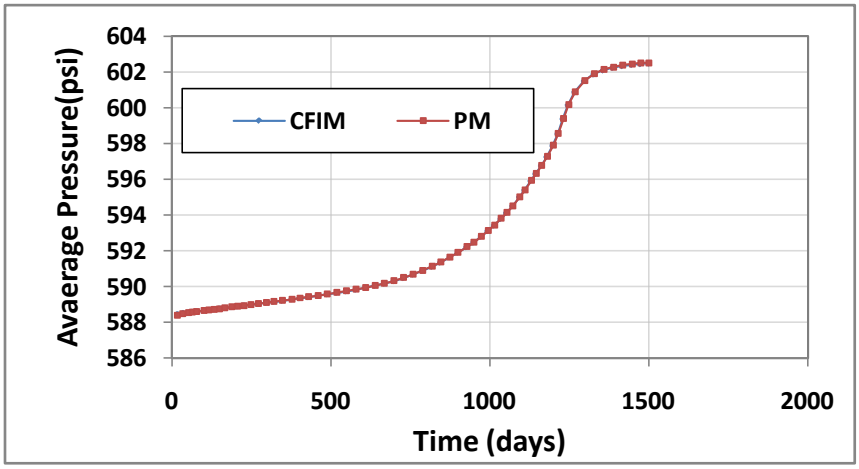

Figure 6.4.2-6: Average reservoir pressure (psi) versus time (days) for Case I.

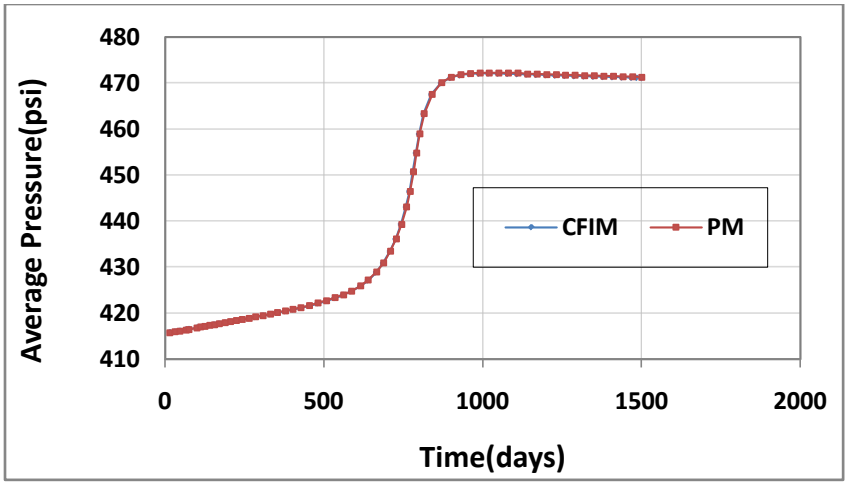

Figure 6.4.2-7: Average reservoir pressure (psi) versus time (days) for Case II.

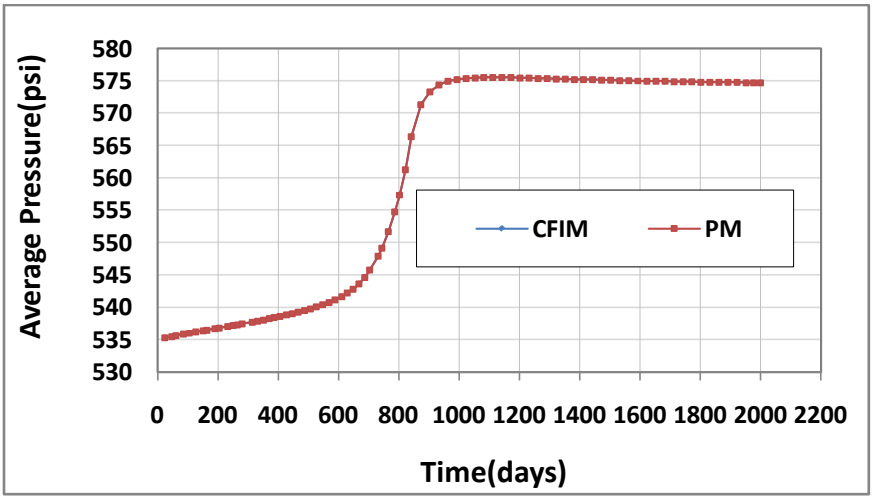

Figure 6.4.2-8: Average reservoir pressure (psi) versus time (days) for Case III.

The third type of validation curves, i.e., the total oil production rate curves for both PM and CFIM, are shown in Figures 6.4-9, 6.4-10, and 6.4-11. The partition method matches the traditional fully implicit method very well.

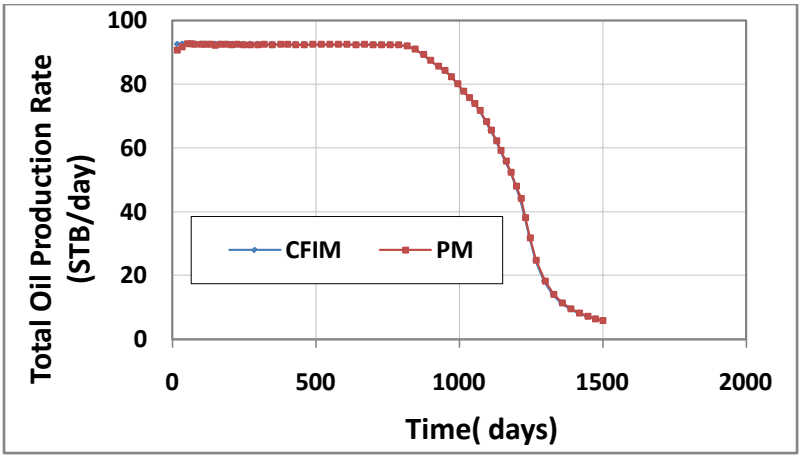

Figure 6.4.2-9: Total oil production rate (STB/day) versus time (day) for Case I.

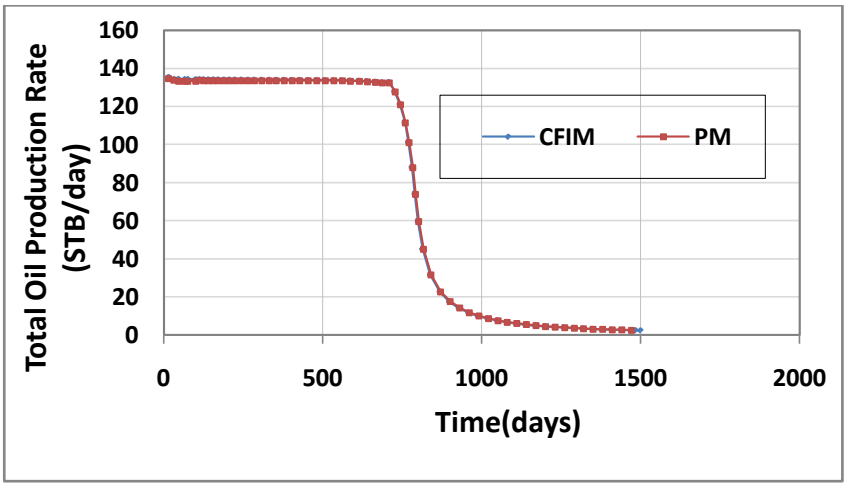

Figure 6.4.2-10: Total oil production rate versus time (days) for Case II.

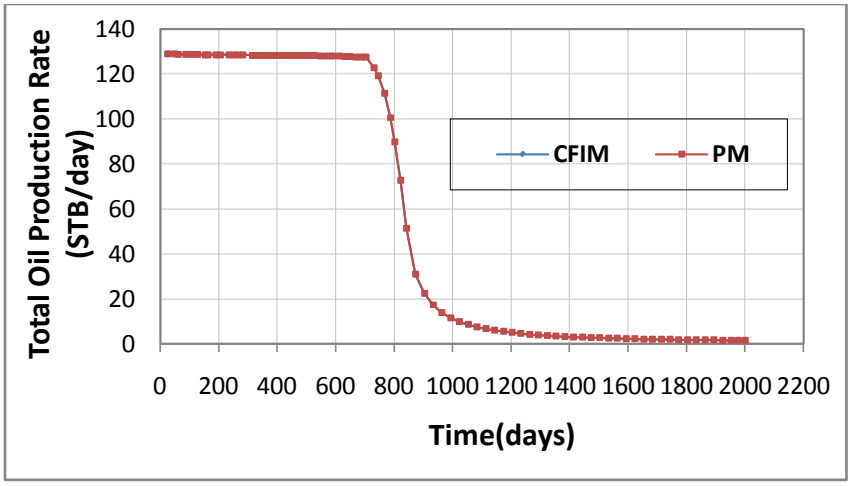

Figure 6.4.2-11: Total oil production rate (STB/day) versus time (days) for Case III.

6.4.3 Comments and Conclusions

A thermal module has been implemented into a GPAS using PM. Three test cases show that PM becomes more efficient than CFIM as the number of gridblocks increased. Three types of validation curves for the three cases show that the results of both approaches are in good agreement.

The results of the numerical tests indicate that although PM required more calculation, the PM can still be about 10% faster than the CFIM with the suitable acceleration techniques for the three test cases while maintaining the same level of accuracy. By increasing the number of gridblocks and components, PM can be much more efficient than CFIM.

Compared to CFIM in implementation, PM does not change the original code drastically, but provides an easier approach of adding new modules into a reservoir simulator.

Chapter 7: Iterative Coupling Methods

In this chapter, three iterative coupling approaches will be discussed.

7.1 Introduction to Iterative Coupling Methods

The iterative coupling method was proposed and described by Settari et al. in 1989 for coupling the fluid and solid mechanics in the new model, representing a significant advance in the realism of modeling of fluid flow and fracturing in oil sands. Then the technical term “iterative coupling” was clearly defined for modular coupling of a commercial reservoir simulator with a 3D stress code (Setterial et al. 1994). The similar modular coupling between a commercial simulator with a 3D stress code and fracture-propagation model was shown by Settari et al. in 1995 and 1998 (Setterial et al.1995, 1998). Then the extension of Setterial et al. in 1998 for modeling compaction, and the application of the model in full-field study was introduced (Setterial et al. 2001). The better relationship between porosity and pressure, temperature, and mean total stress was developed for increasing the accuracy of coupling and decreasing the number of the coupling iterations (Tran et al. 2002, 2004). Thomas et al. 2003 presented an iterative procedure to couple geomechanics and reservoir simulation of weak rock formations with complex constitutive behavior. The improvement of porosity formula for an iterative coupling was described in 2005 (Tran et al. 2005). In an overview of iterative coupling methods (Tran et al. 2005), the iterative coupling method is described in detail from the basic equations for geomechanics and reservoir flow to the coupling technique through porosity and permeability. The practical iterative coupling of geomechnaics with reservoir simulation for field scale was presented in 2007 (Samier et al. 2007, 2008). Lu et al. 2007 applied an iterative coupling method to both the pressure equation and

saturation/concentration equations for multiphase flow to compare with the fully implicit method and IMPES method. The main difference between an iterative coupling in the above references and PM is that in the iteratively coupled method the information is passed back and forth explicitly between reservoir simulator and a new coupled module, whereas the new unknowns in PM such as temperature is solved one time from the Schur complement. Therefore, original unknowns are passed from the original simulator to a new coupled module only one time. The solving procedures are so complicated that they guarantee that PM has the same convergence as that of CFIM. Iterative coupling methods mentioned in the references are easier to implement than PM.

7.2 Procedure of Iterative Coupling Methods

7.2.1 Iterative Coupling Methods in Literatures

Settari's iterative coupling method (SICM) is described in Figure 7.2.1-1 (Settari et al. 1994). There are interactions between geomechanical behavior and multiphase flow and heat transfer in porous media in the iterative coupling procedure.
The convergence criteria is that $\left\| P^{(k+1)} - P^{(k)} \right\|_2$ reaches the given the tolerance.

1) Calculate porosity ϕ^* in the reservoir model for any P and T.
2) Obtain new iterate of solution $P^{(k+1)}$ and $T^{(k+1)}$ from reservoir simulator, transfer them to the stress model, and solve stress equations for new iterate of $\sigma_{x,y,z}$ and $\varepsilon_{x,y,z}$.
3) Calculate true porosity $\phi^{(k+1)}$.
4) Update the coefficients related with $\phi^{(k+1)}$.
5) Calculate new volumetric strain and reservoir porosity.
6) Check the convergence by taking $\left\| P^{(k+1)} - P^{(k)} \right\|_2$ to see if converged. If not, go back to step 1.

Figure 7.2.1-1: Flowchart of SICM (Settarial et al. 1994).

Tran et al.'s iterative coupling method (TICM) was described as that "reservoir flow variables and geomechanics variables are solved separately and sequentially by a reservoir simulator and a geomechanics module, and the coupling terms are iterated on each time step" (Tran et al. 2002 page 2). The coupling iteration is controlled by a convergence criterion that is normally based on either pressure or stress changes between

the last two iterates of the solution. The Flowchart of the iterative coupling is shown in Figure 7.2.1-2.

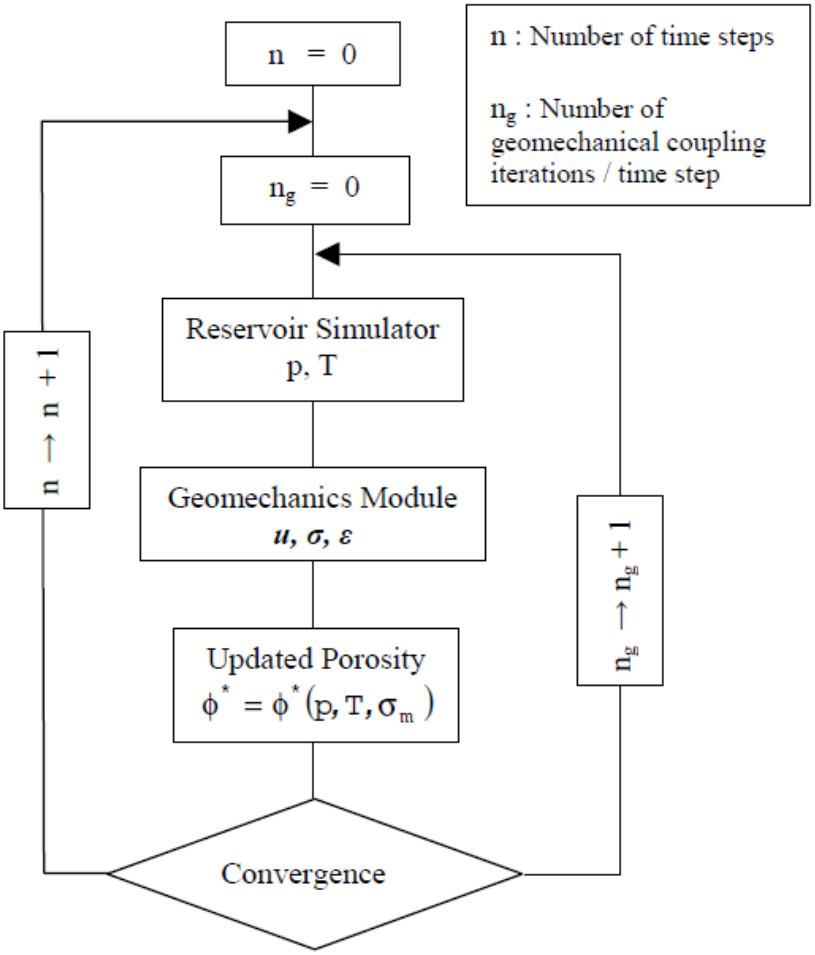

Figure 7.2.1-2: Flowchart of TICM (Tran et al. 2002).

Chin et al. 2002 proposed an iterative, fully coupled procedure that integrates reservoir simulation with geomechanics in a generalized fashion. Both the reservoir multiphase flow model and the geomechanics model are linked together based on discretized coupled equations. The shared variables are porosity and fluid pressure for each gridblock. On each time step, both models iteratively exchange numerical values of

the needed variables on every gridblock basis. The geomechanics model provides state variables of the reservoir model, porosity, and its derivative with respect to pressure. The iteratively of solving reservoir model and geomechanics model and passing the needed solution of variables between two modules continues until convergences criteria are reached. The flowchart of Chin's iteratively, fully coupled method (CIFCM) is presented in Figure 7.2.1-3.

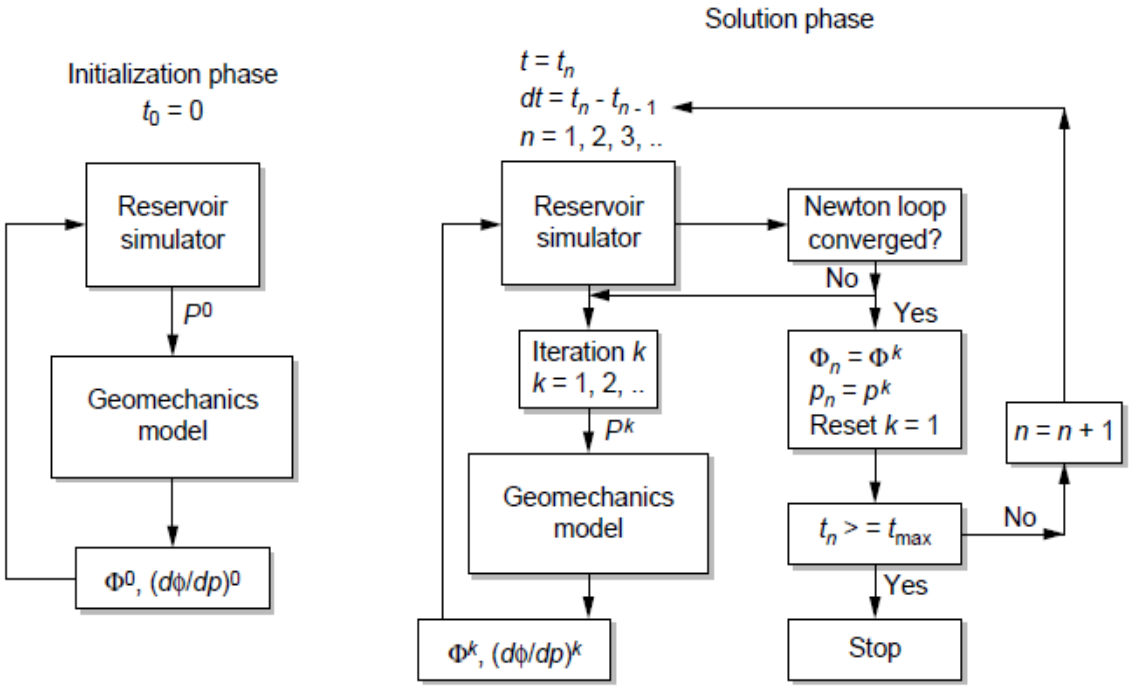

Figure 7.2.1-3: Flowchart of CIFCM (Chin et al. 2002).

Lu et al. applied iterative coupling to multiphase flow simulation (Lu et al. 2007). The primary variables were chosen as water pressure P_w and water saturations S_w for a two-phase oil/water model. Summing water and oil mass conservation equations resulted in water pressure equation. Then Lu's iterative coupling procedure (LICP) of solving water pressure P_w and water saturations S_w iteratively until the globe mass balance criteria is reached as described in Figure 7.2.1-4.

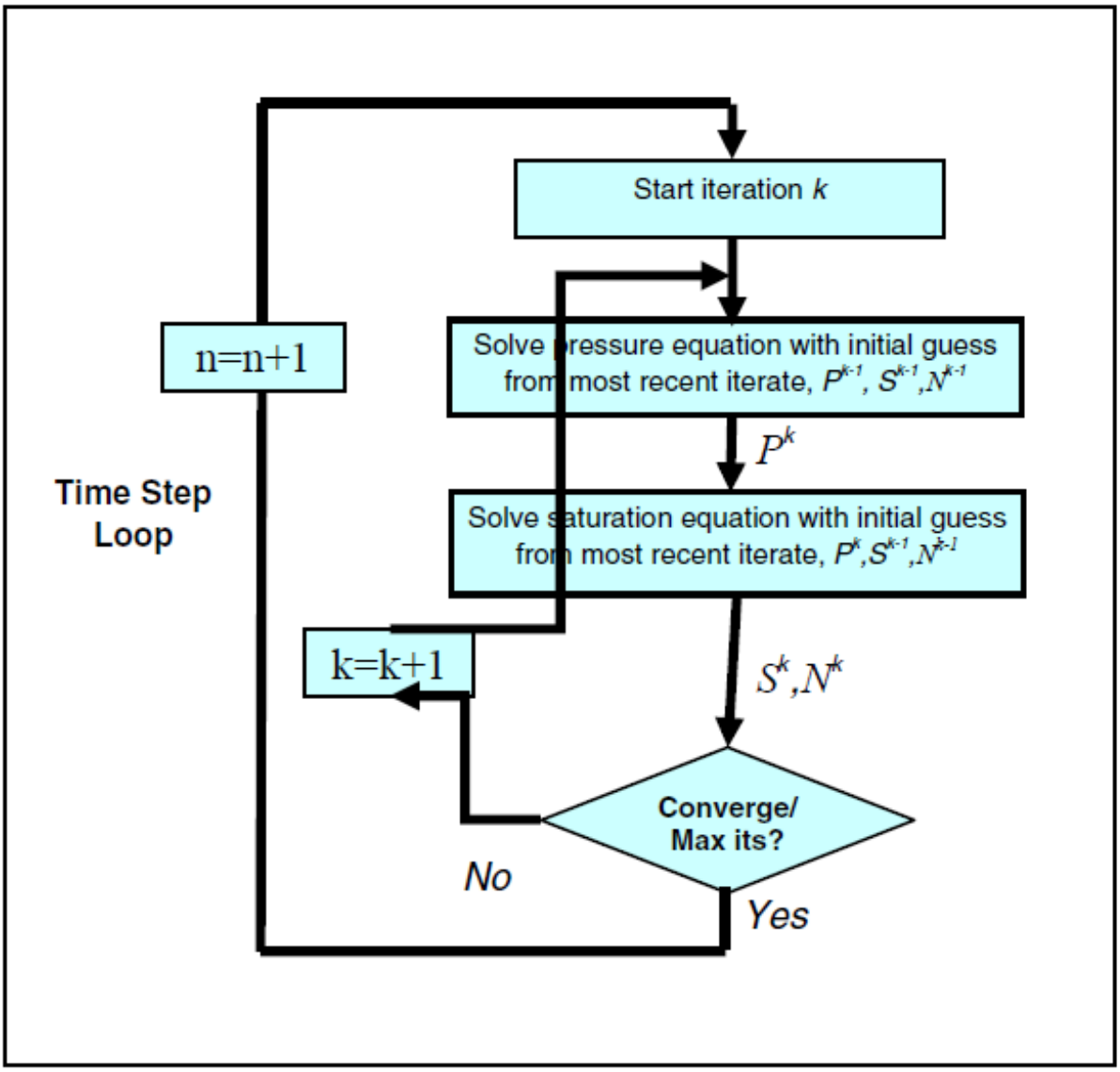

Figure 7.2.1-4: Flowchart of LICP (Lu et al. 2007).

Samier's reservoir simulation coupled with geomechanics has been increasing in recent years as its utility in modeling physical phenomena such as compaction, subsidence, induced fracturing, enhancement of natural fractures, and/or fault activation, and steam-assisted gravity drainage (SAGD) recovery (Samier et al. 2008). Samier's classic iterative coupling method (SCICM) is depicted in Figure 7.2.1-5. Samier's

practical iterative coupling method (SPICM) was presented in Figure 7.2.1-6, which was a loose coupled approach between a reservoir simulator and a geomechanical simulator.

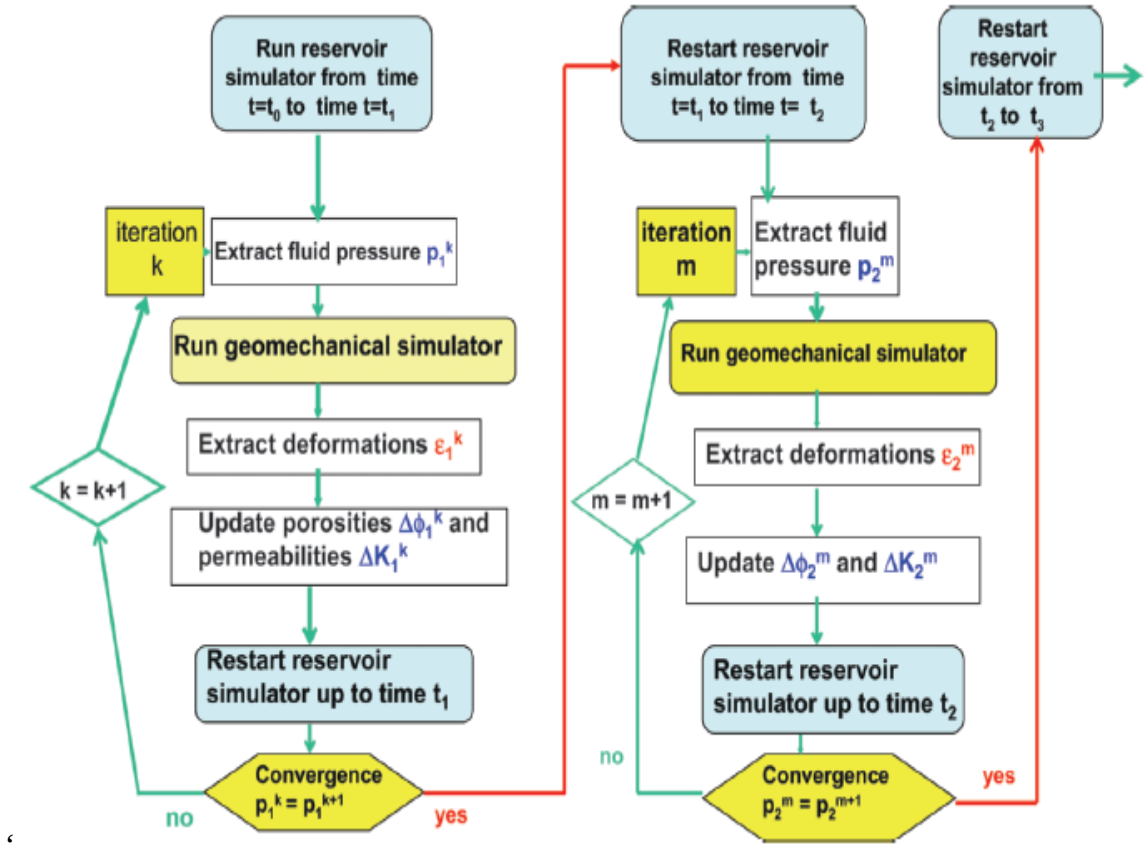

Figure 7.2.1-5: Flowchart of SCICM (Samier et al. 2008).

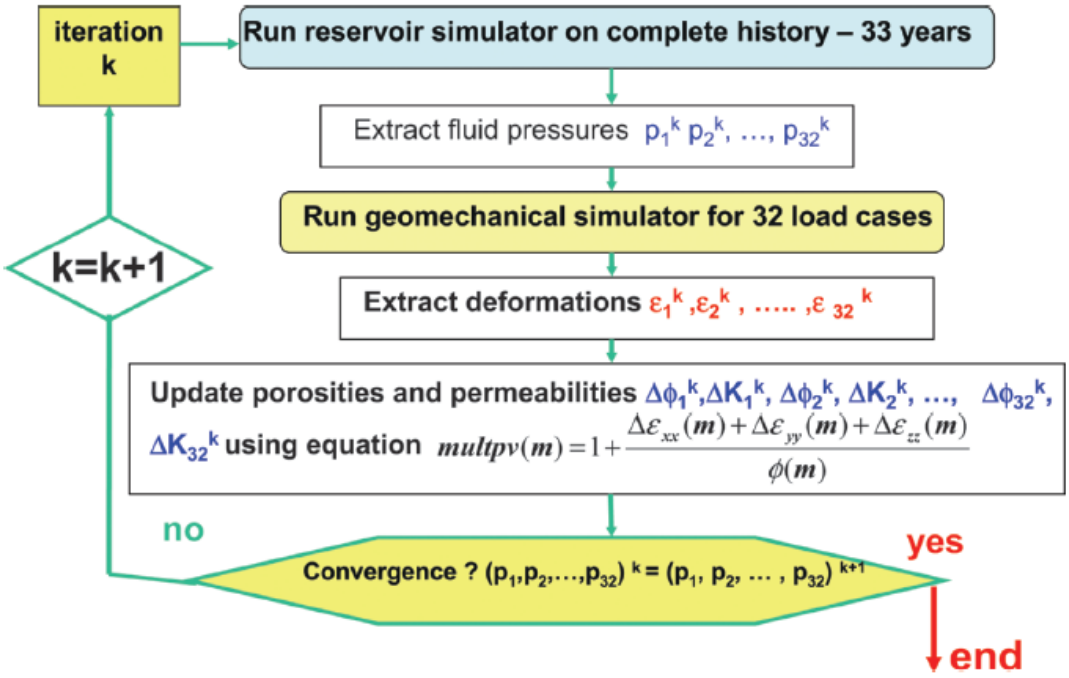

Figure 7.2.1-6: Flowchart of SPICM (Samier et al. 2008).

7.2.2 Iterative Coupled Thermal Module with GPAS

Based on the iterative coupling methods reviewed in Section 7.2.1. Three iterative coupling methods will be proposed and implemented in between GPAS and thermal module.

General, iterative coupling method (GICM): This method was generated based on SICM and TICM. At each time step, the GPAS' variables pressure, concentration and $\ln k$ and thermal module's variable temperature are solved separately and sequentially by GPAS simulator and thermal module. The convergence of criteria is only based on the convergence of GPAS variables. Figure 7.2.2-1 and Figure 7.2.2-2 show the implementation of a flowchart of the GICM.

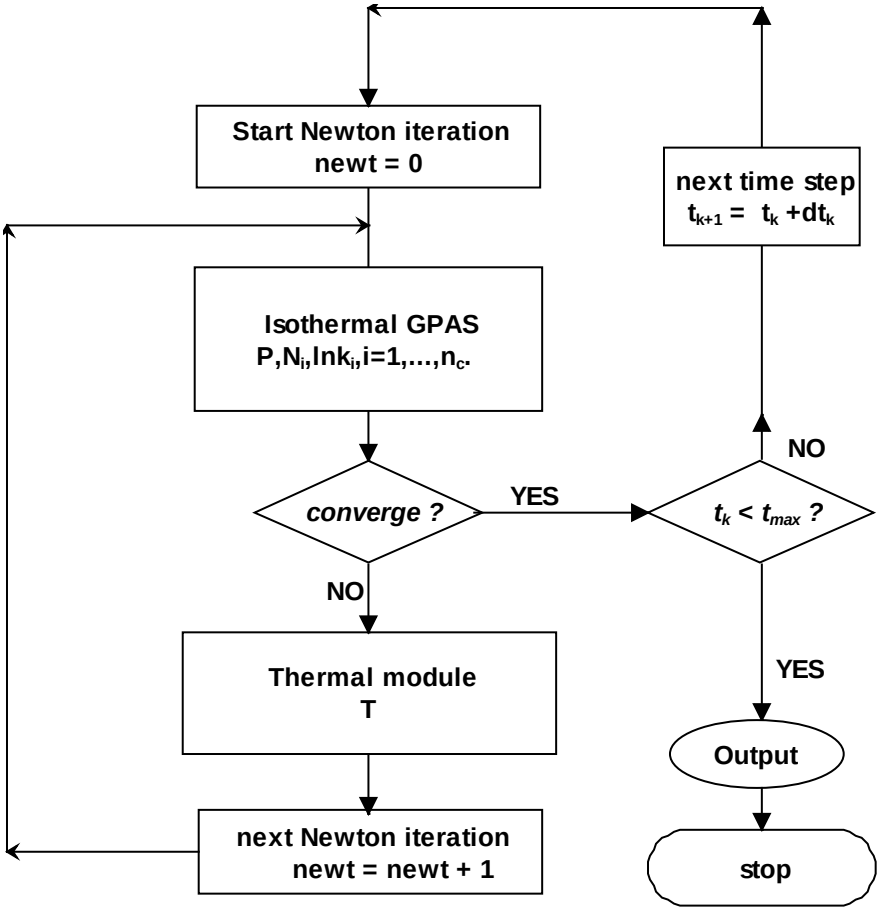

Figure 7.2.2-1: Flowchart of GICM.

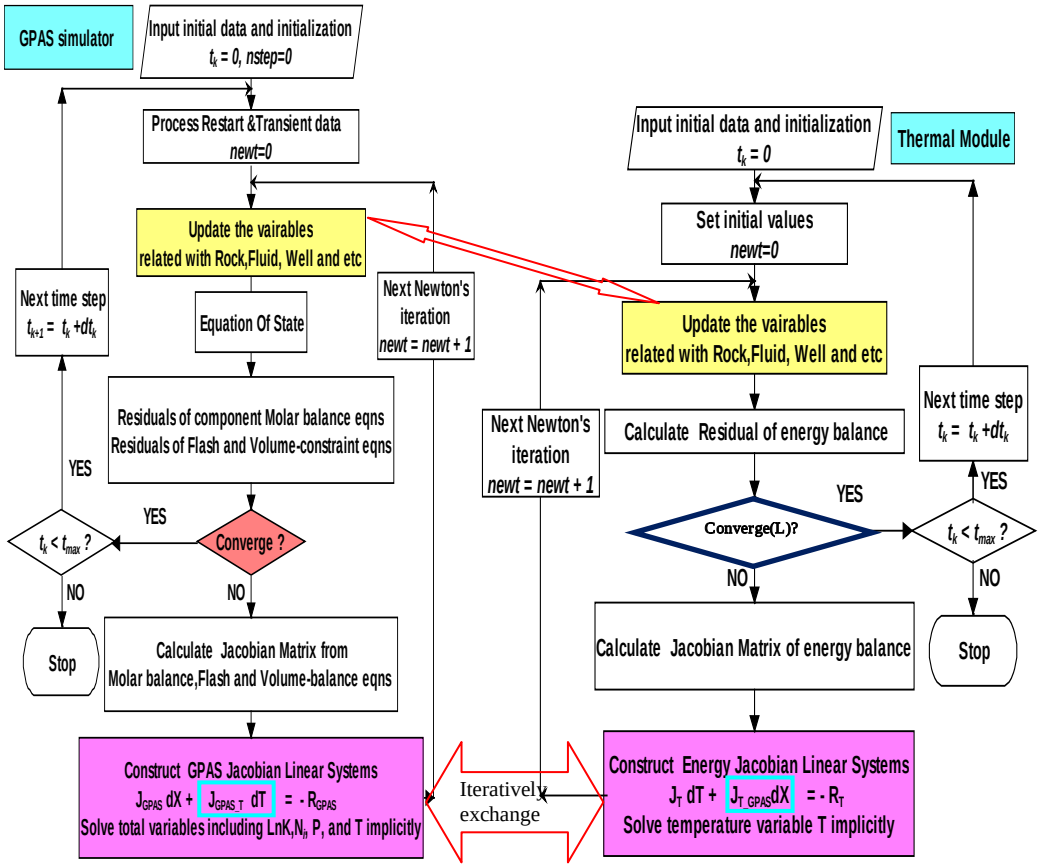

Figure 7.2.2-2: Flowchart of GICM in details.

Fully, iterative coupling method (FICM): This method was generated based on both CFICM and a PM. At each time step, the GPAS' variables pressure, concentration and $\ln k$ and thermal module's variable temperature are solved separately and sequentially by GPAS simulator and thermal module until the variables of GPAS is converged. Figure 7.2.2-3 shows the implementation of flowchart of FICM.

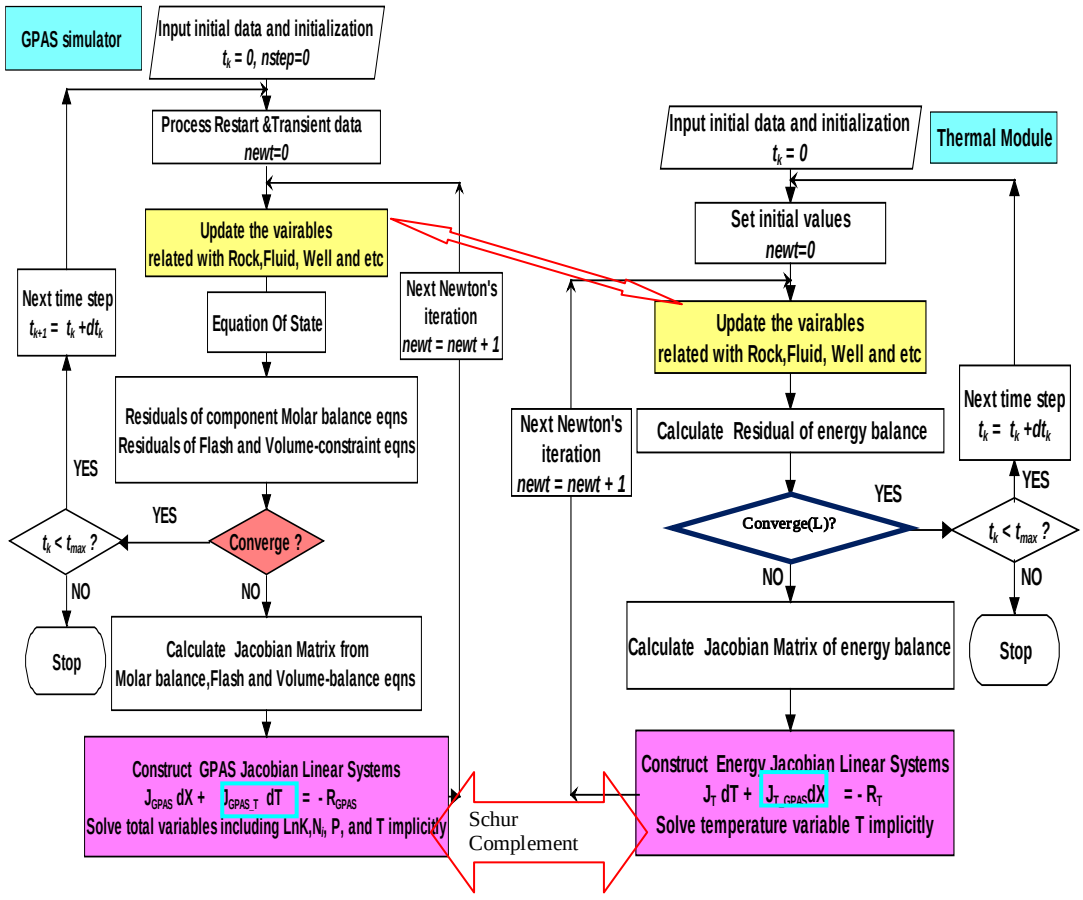

Figure 7.2.2-3: Flowchart of FICM in details.

Loose, iterative coupling method (LICM): This method was generated based on SPICM. At different time steps, the GPAS' variables such as pressure, concentration, and $\ln k$ and thermal module's variable temperature are solved separately and sequentially by the GPAS simulator and thermal module until variables of GPAS are converged. Figure 7.2.2-4 shows the implementation of a flowchart of LICM.

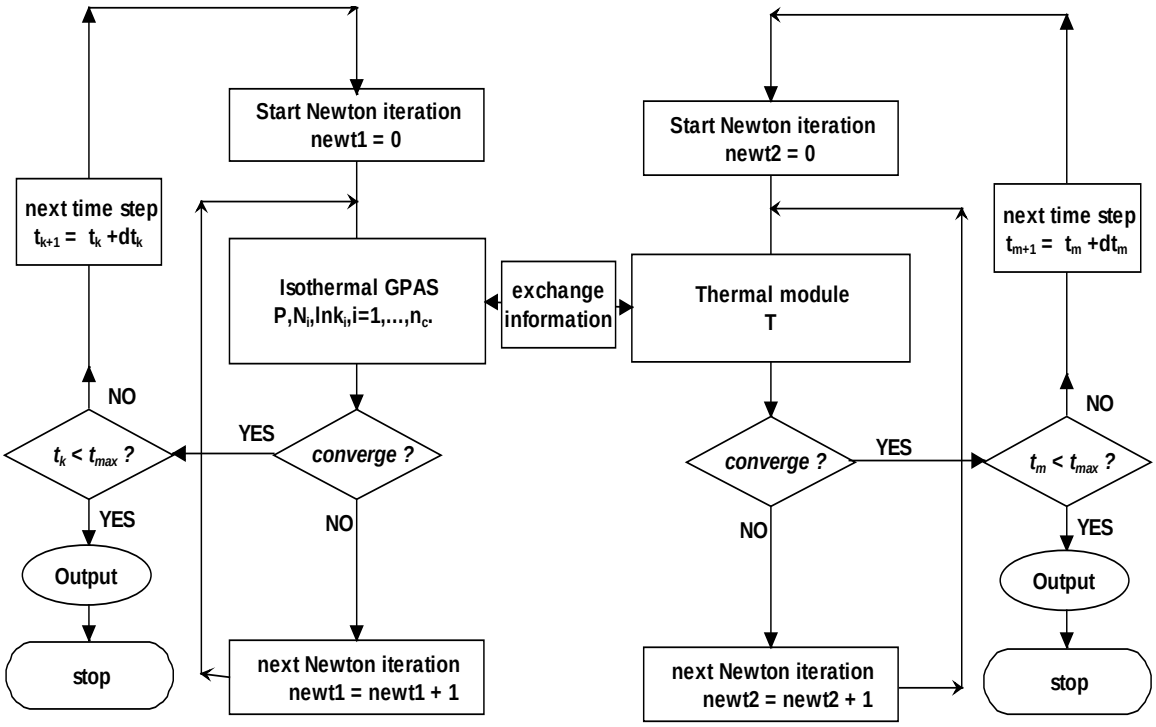

Figure 7.2.2-4: Flowchart of LICM.

7.3 Numerical Results

In this section, we present the results of two iteratively coupling methods, i.e., FICM and GICM instead of LICM due to a lack of LICM convergence.

7.3.1 Two-Component Water Flooding with 10x10x5 Gridblocks (Case I)

This test includes two components, water and $C_{17,}$ in a 3D homogeneous reservoir 480x480x50 ft^3 in size with a permeability of 250 md and a porosity of 0.3. The initial pressure and temperature of reservoir are 600 psi and 240 ^oF, respectively. The heat capacity ratio for fluid and rock is one. The volumetric heat capacity is 35 Btu/(ft^3-°R) and the thermal conductivity is 35 Btu/(ft-day-°R). The reservoir is divided into 10x10x5 gridblocks. Steam hot water with the rate of 1500 lbmol/day and temperature of 790 °F is injected into the reservoir. The production bottomhole pressure is maintained at 580 psia. The simulation period is 1500 days. The reservoir region with the location of wells ("Yellow" is a production well, "blue" is an injection well) of Case I is shown in Figure 7.3.1-1.

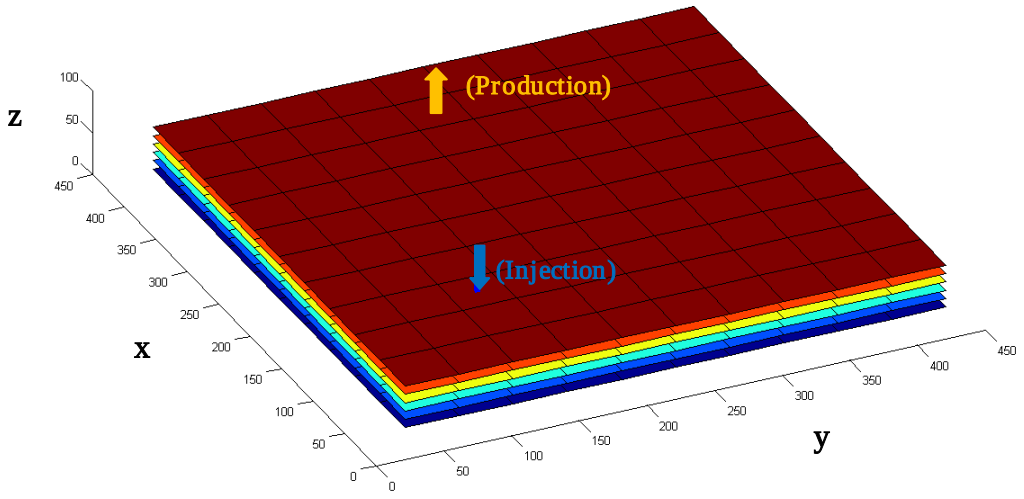

Figure 7.3.1-1: Reservoir model for Case I.

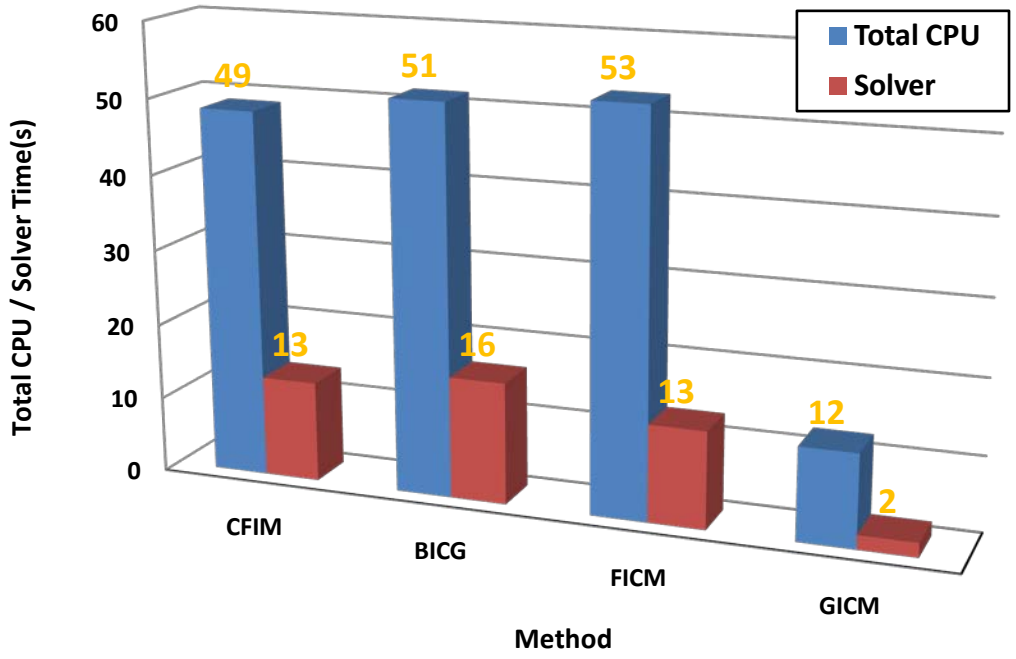

Figure 7.3.1-2: Computational time for Case I.

Figure 7.3.1-2 shows the results of computational times for four methods using GPAS for Case I. has the same solver's running time of CFIM, GICM is much faster than CFIM, which only takes one-sixth of the running time of CFIM or FICM.

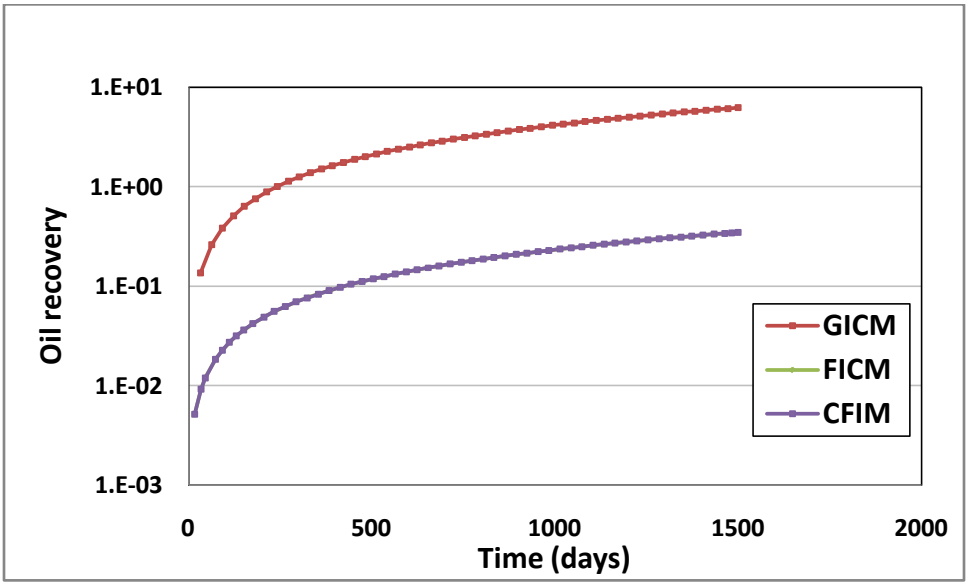

Figure 7.3.1-3: Oil recovery curve vs. time.

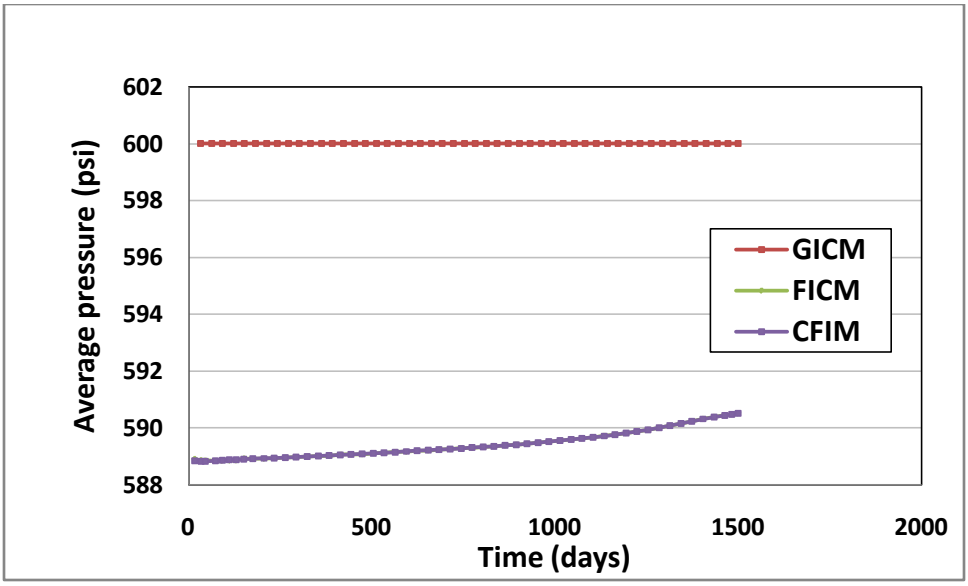

Figure 7.3.1-4: Average reservoir pressure (psi) vs. time.

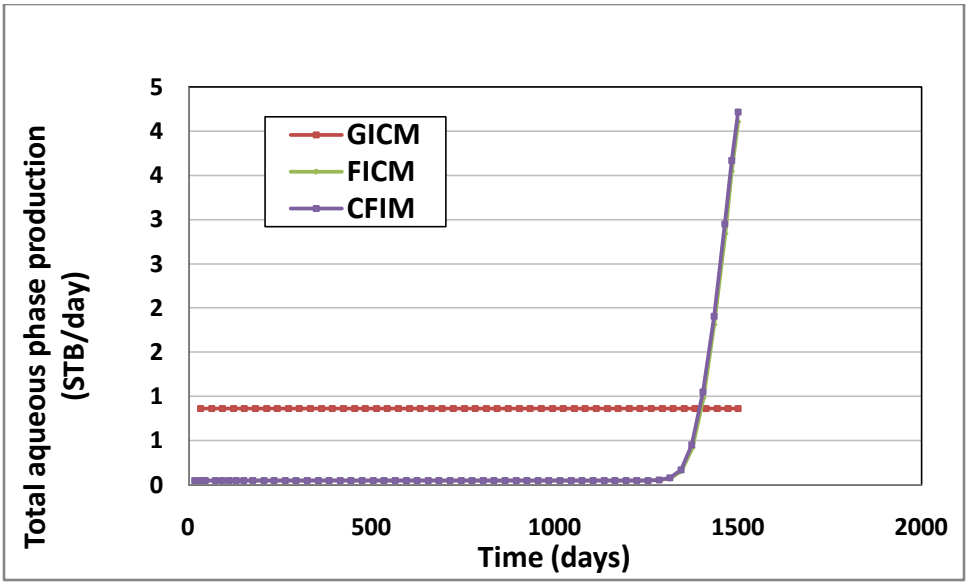

Figure 7.3.1-5: Total aqueous phase production (STB/day) vs. time.

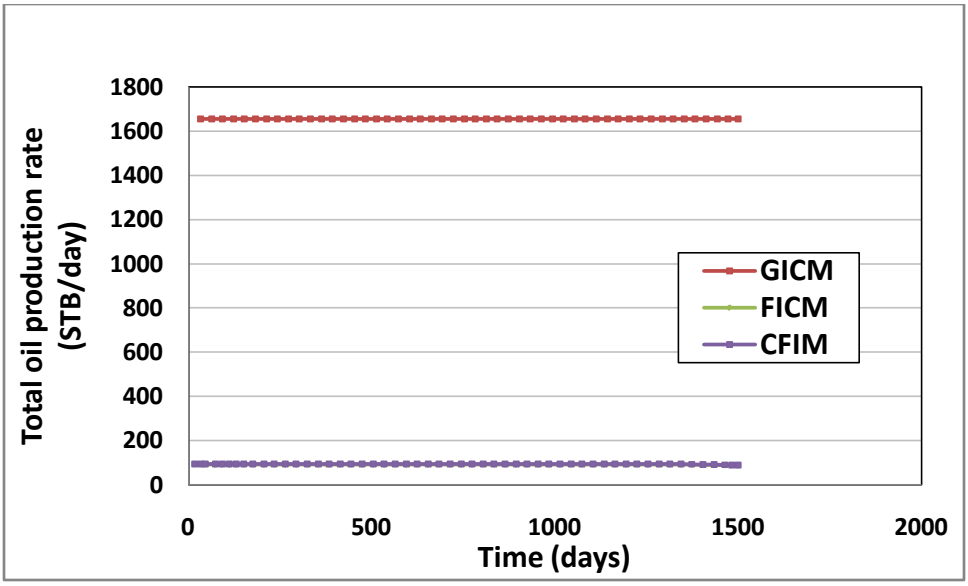

Figure 7.3.1-6: Total oil production rate (STB/day) vs. time.

Figures 7.3.1-3 through 7.3.1-6 show validation results for the three methods. The figures show that the three methods have good agreement with the average of pressure curves. However, for other curves such as oil recovery curve, total aqueous phase production rate curve, and total oil production curve, FICM matched CFIM very well, but GICM did not match CFIM.

7.3.2 Two-Component Water Flooding with 20x20x5 Gridblocks (Case II)

This test is the same as Case I, but we use a larger number of gridblocks for this study. The reservoir is divided into 20x20x5 gridblocks. Hot water with the rate of 1500 Lbmol/day and temperature of 790 °F is injected into the reservoir. The production bottomhole pressure is maintained at 580 psia. The simulation period is 1500 days. The reservoir region with the special location of wells ("Yellow" is a production well, "blue" is an injection well) of Case II is shown in Figure 7.3.2-1.

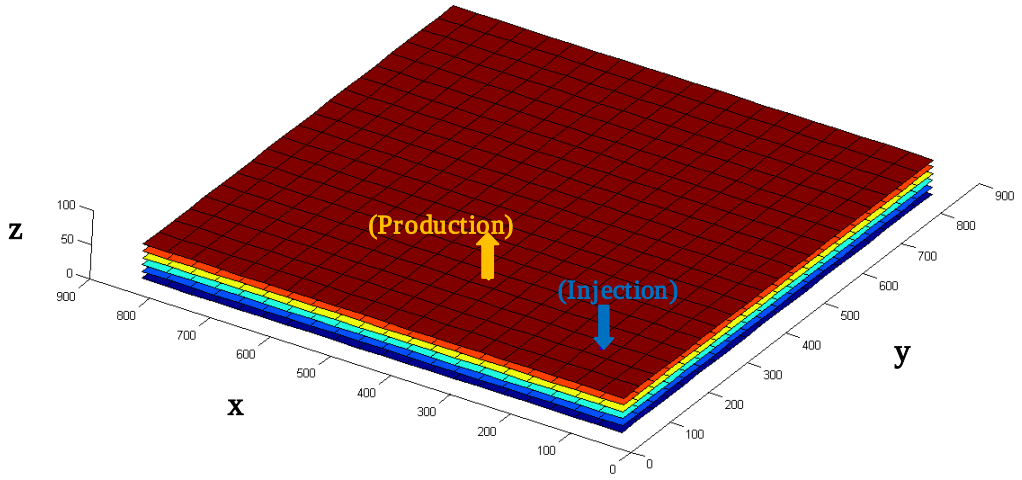

Figure 7.3.2-1: Reservoir model for Case II.

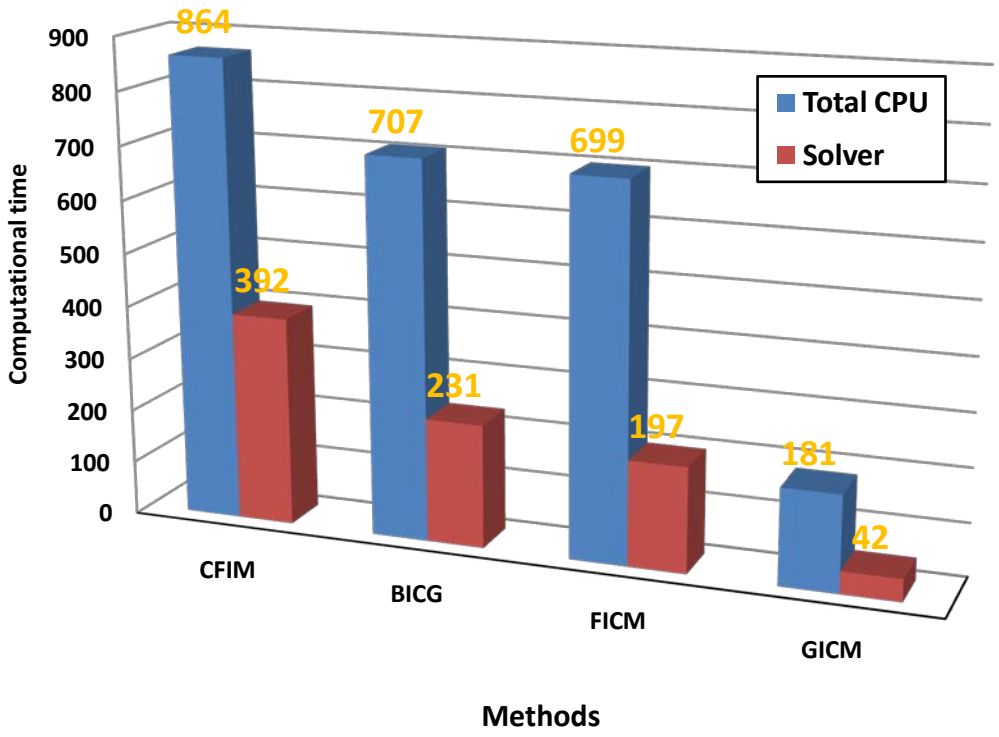

Figure 7.3.2-2: Computational time for Case II.

Figure 7.3.2-2 shows the computational results for Case II. FICM is two times faster than CFIM. GICM is the fastest among the four methods; GICM is 9.3 times faster than CFIM, and 4.6 times faster than CFIM.

Figures 7.3.2-3 through 7.3.2-6 show the validation results for the three methods. The figures show FICM matches CFICM very well for all validation curves. GICM matches CFICM very well for the average reservoir pressure curve, but not well for the rest of the validation curves.

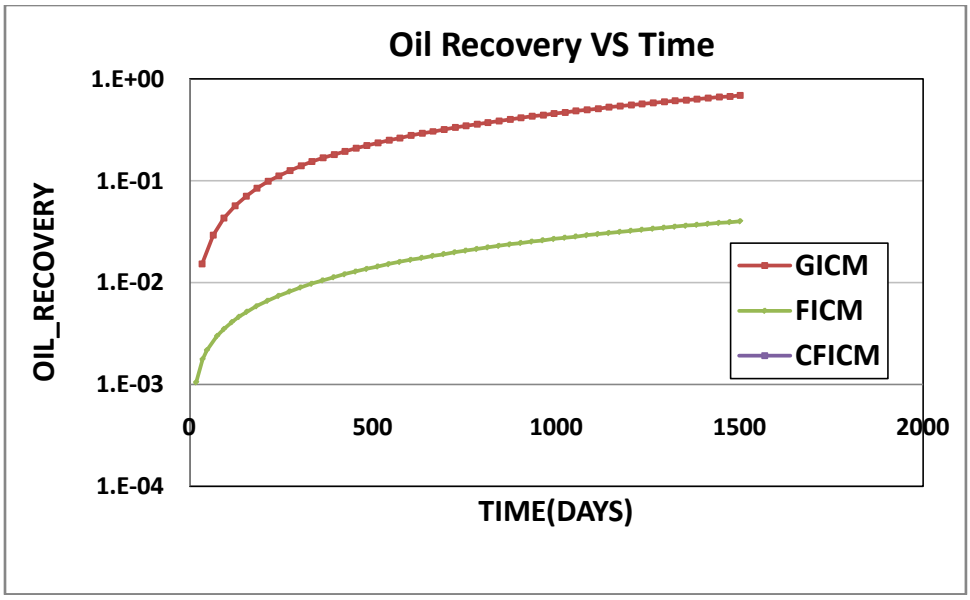

Figure 7.3.2-3: Oil recovery curve vs. time.

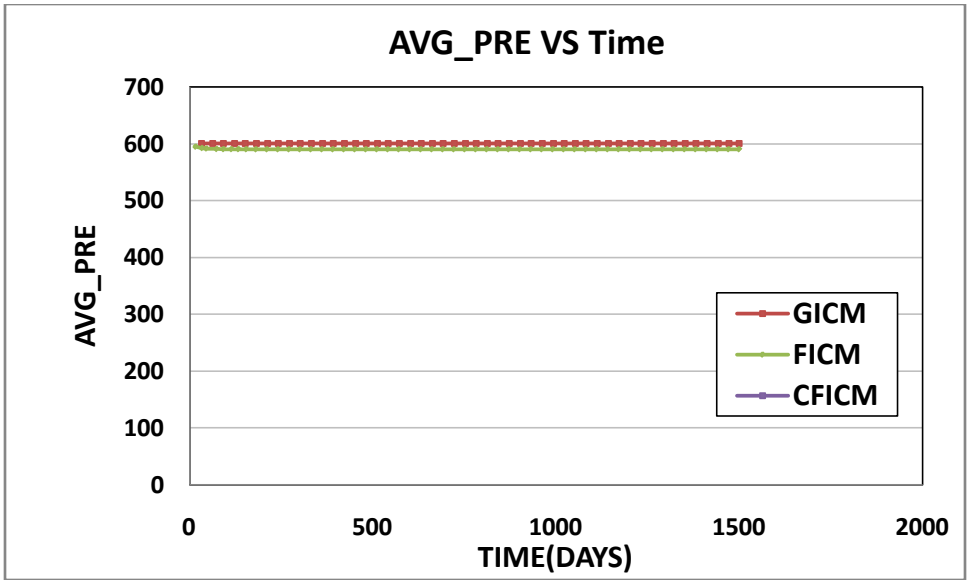

Figure 7.3.2-4: Average pressure (psi) vs. time.

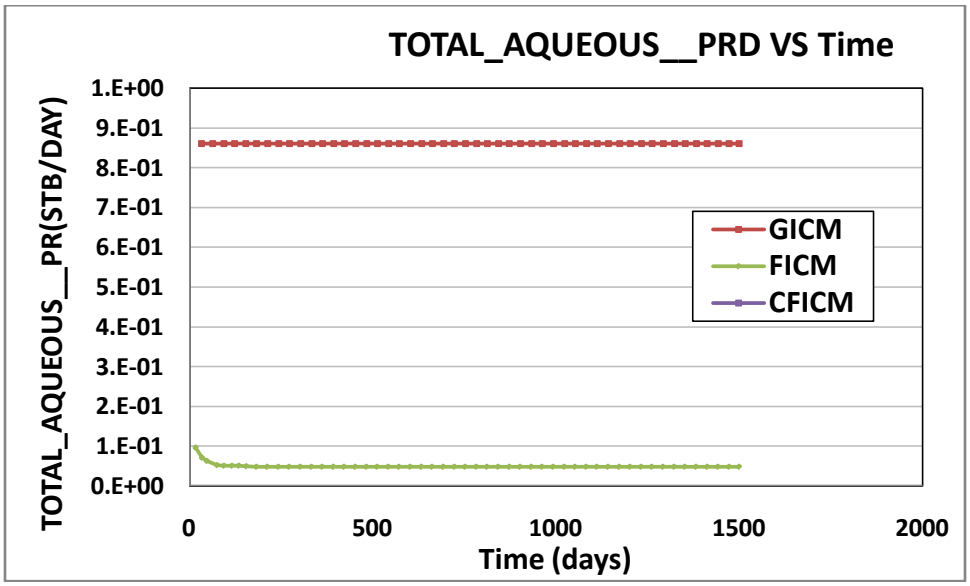

Figure 7.3.2-5: Total aqueous phase production (STB/day) vs. time.

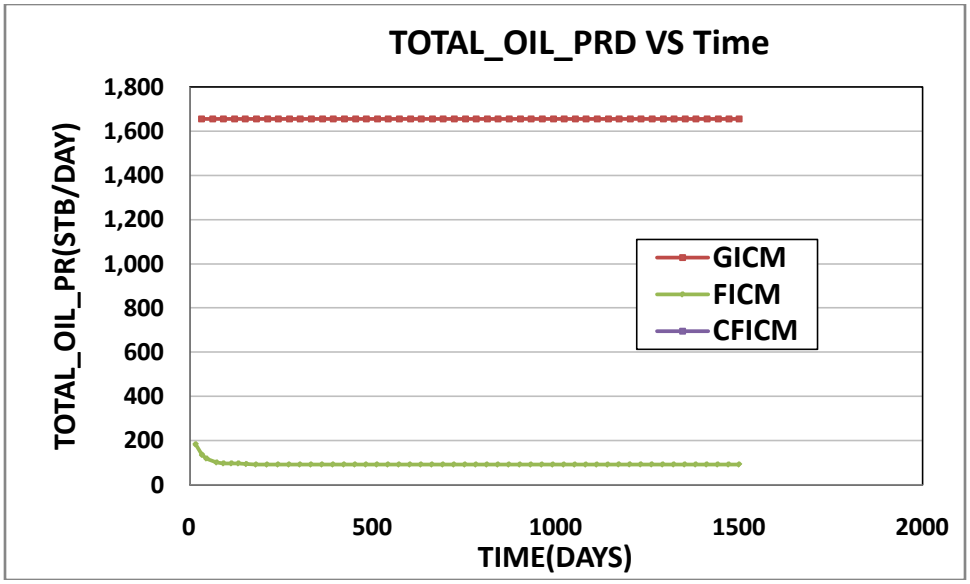

Figure 7.3.2-6: Total oil production (STB/day) vs. time.

7.3.3 Three-Component Water Flooding with 12x12x5 Gridblocks (Case III)

Case III is a four-component case consisting of C_{17}, C_{19} and water components. The 3D homogeneous reservoir is 480x480x50 ft^3 in size with a permeability of 250 md and porosity of 0.3. The initial pressure and temperature of reservoir are 600 psi and 790 ^oF, respectively. The heat capacity ratio for fluid and rock is one. The volumetric heat capacity is 35 Btu/(ft^3-°R) and the thermal conductivity is 35 Btu/(ft-day-°R). The reservoir is divided into 12x12x5 gridblocks. Hot water at the rate of 1500 Lbmol/day and temperature of 790 °F is injected into the reservoir. The production bottomhole pressure is maintained at 580 psia. The simulation period is 1500 days. The reservoir region with the location of wells ("Yellow" is a production well, "blue" is an injection well) of Case III is shown in Figure 7.3.3-1.

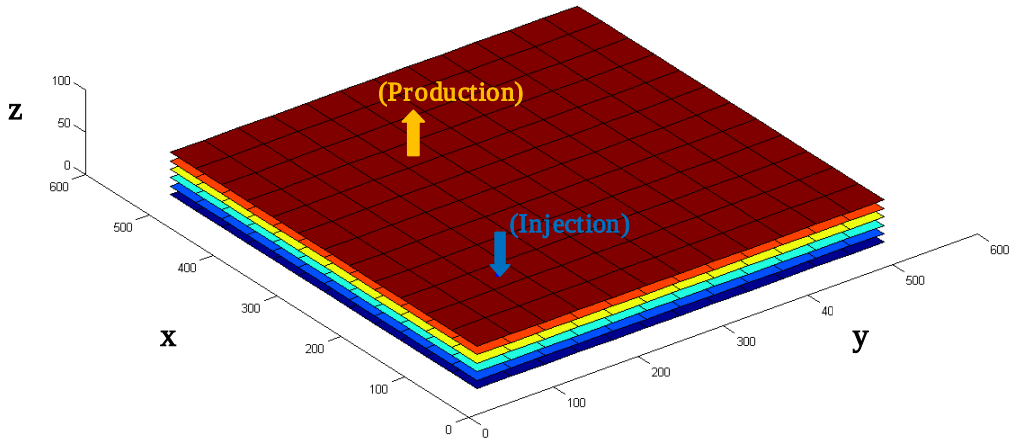

Figure 7.3.3-1: Reservoir model for Case III.

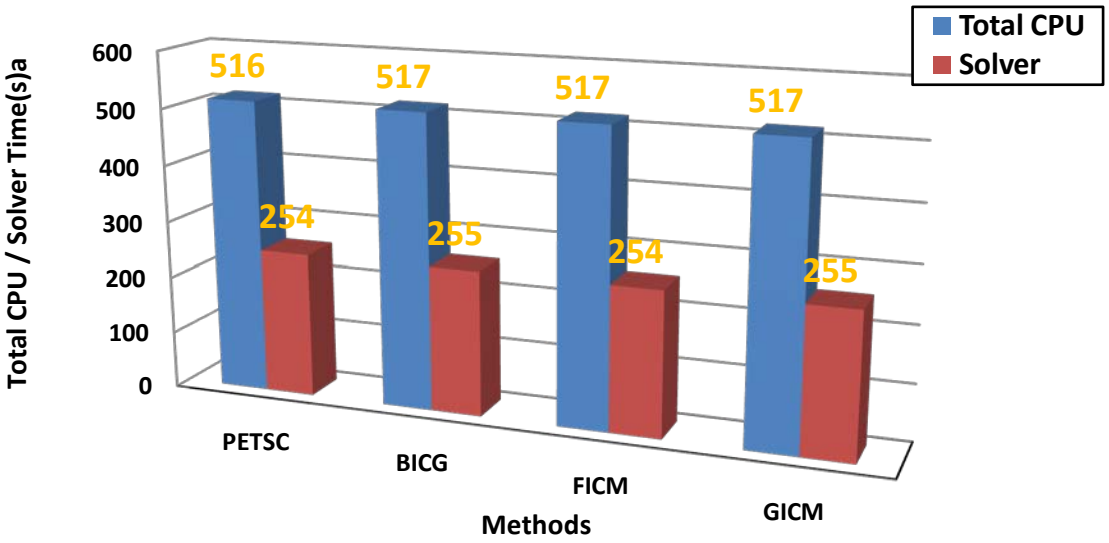

Figure 7.3.3-2: Computational time for Case III.

Figure 7.3.3-2 shows the computational results for Case III. FICM is as fast as CFIM. GICM is a little slower than FICM. All four methods have almost the same computational times.

Figures 7.3.3-3 through 7.3.3-5 show the validation results of three methods. The figures show all methods match CFICM very well for all validation curves. Therefore, GICM can also solve this case very well.

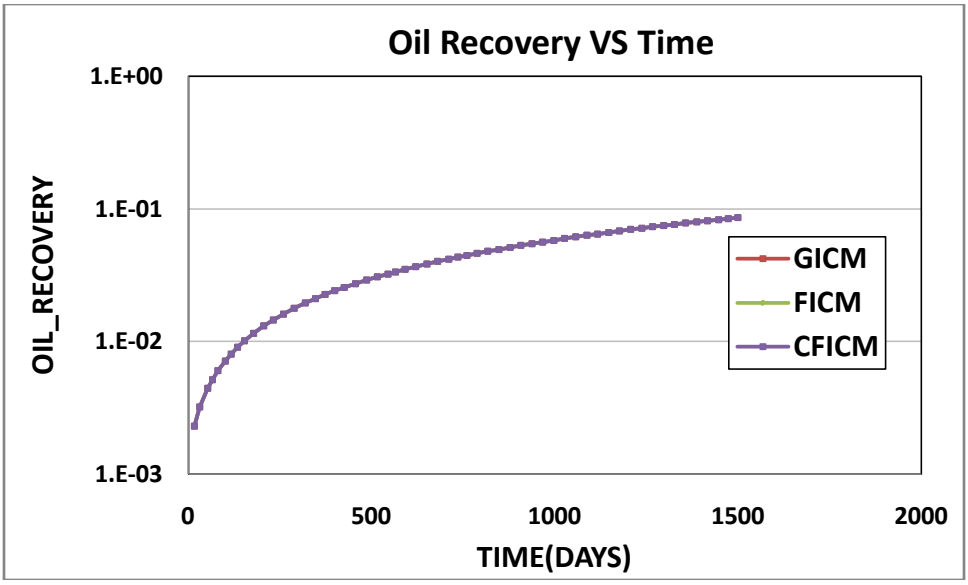

Figure 7.3.3-3: Oil recovery curve vs. time.

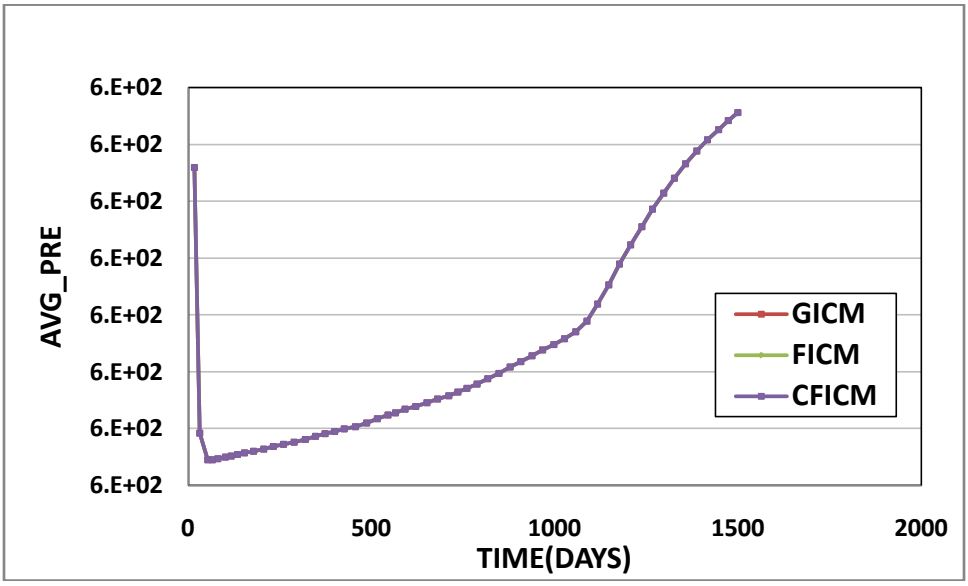

Figure 7.3.3-4: Average pressure (psi) vs. time.

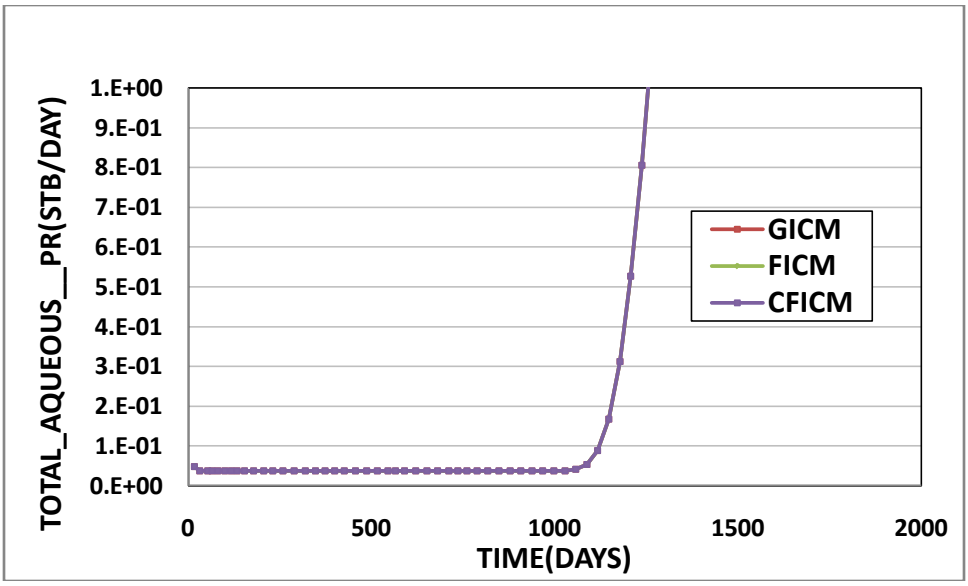

Figure 7.3.3-5: Total aqueous phase production (STB/day) vs. time.

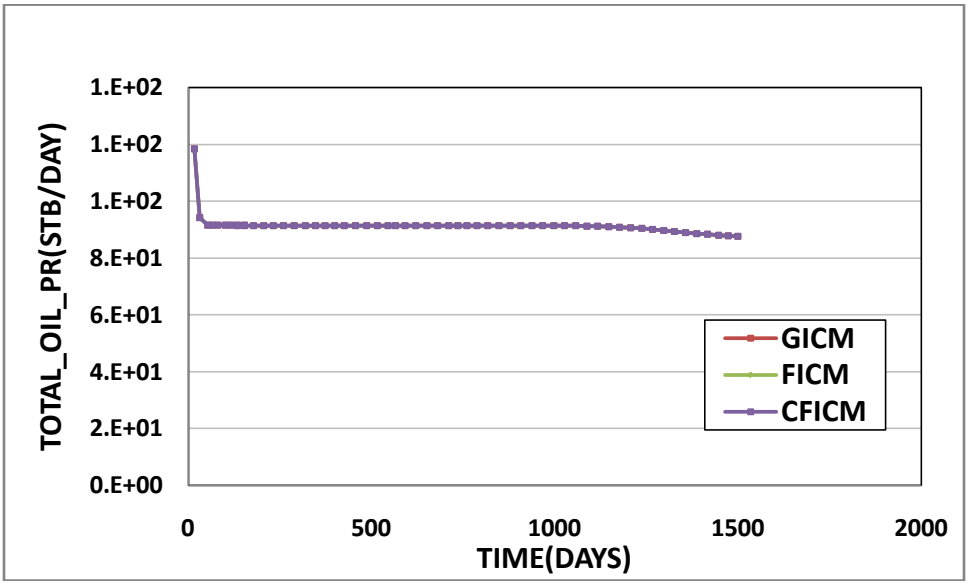

Figure 7.3.3-6: Total oil production (STB/day) vs. time.

7.3.4 Four-Component Water Flooding with 18x18x3 Gridblocks (Case IV)

Case IV is a four-component case consisting of C_6, C_{17}, C_{19} and water components. The 3D homogeneous reservoir is 720x720x30 ft^3 in size with a permeability of 150 md and porosity of 0.3. The initial pressure and temperature of reservoir are 450 psi and 240 ^oF, respectively. The heat capacity ratio for fluid and rock is one. The volumetric heat capacity is 35 Btu/(ft^3-°R) and the thermal conductivity is 35 Btu/(ft-day-°R). The reservoir is divided into 16x16x3 gridblocks. Steam water at the rate of 2200 Lbmol/day and temperature of 790 °F is injected into the reservoir. The production bottomhole pressure is maintained at 400 psia. The simulation period is 1500 days. The reservoir region with the locations of two wells ("Yellow" is a production well, "blue" is an injection well) of Case IV is shown in Figure 7.3.4-1.

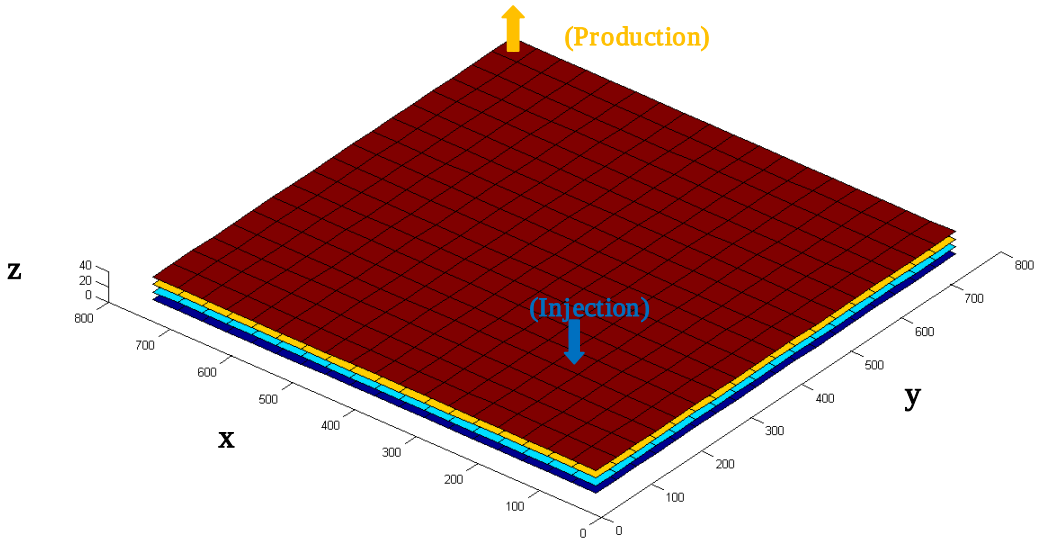

Figure 7.3.4-1: Reservoir model for Case IV.

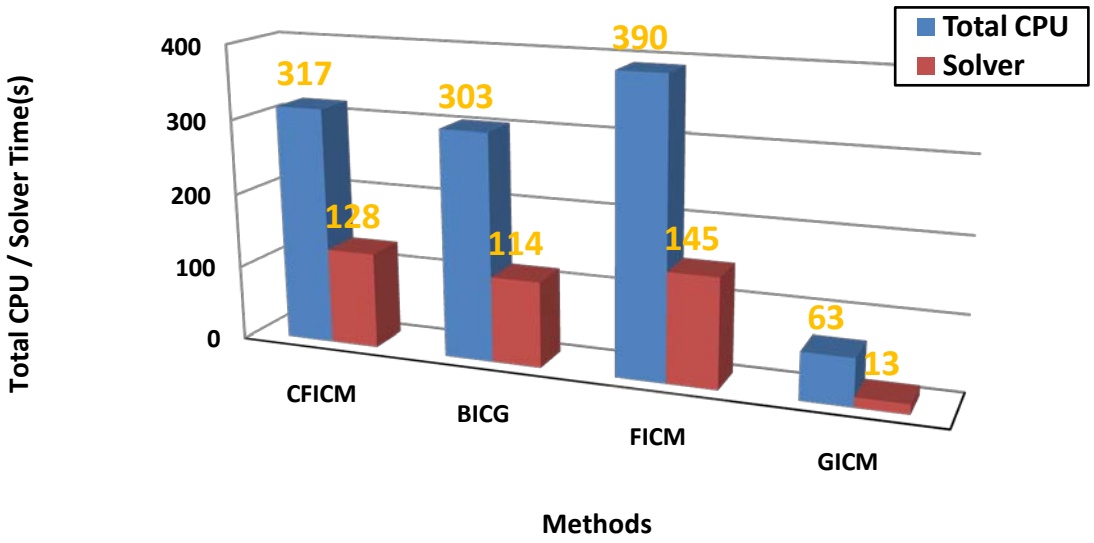

Figure 7.3.4-2: Computational time of Case IV.

Figure 7.3.4-2 shows the CPU times for Case IV using GPAS for four methods: CFICM, BICG, FICM, and GICM. The solver time of GICM is less than 49% of that of CFICM, less than 43% of that of FICM, and less than 55% of that of FICM. Therefore, Figure 7.3.4-2 shows that GICM is the fastest method, and that FICM is slower than CFICM.

Figures from 7.3.4-3 through 7.3.4-6 show validation results of the three methods. The figures show that CFICM matches with FICM well, and GICM matches FICM very for the average reservoir pressure and total aqueous phase production rate validation curve, but not well for the rest of the curves.

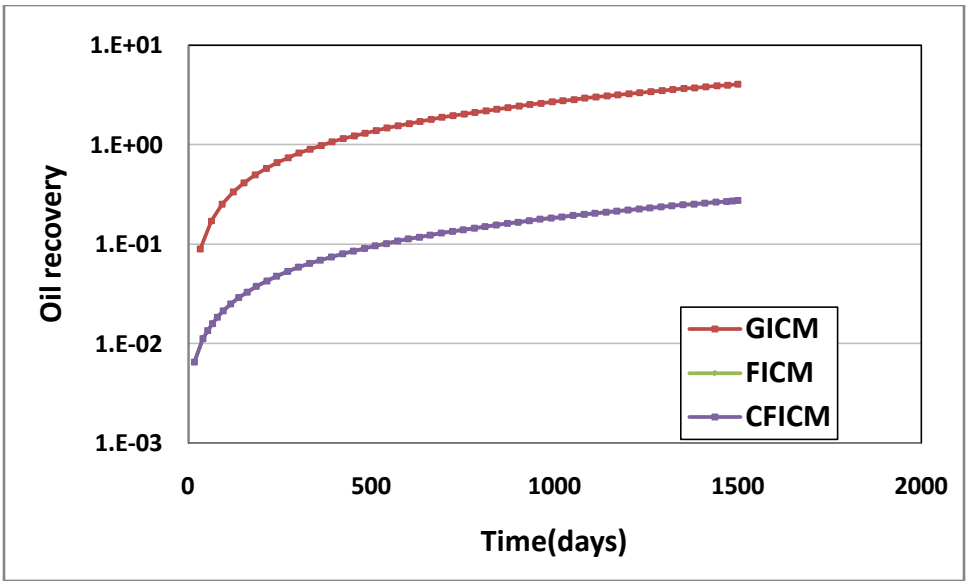

Figure 7.3.4-3: Oil recovery curve.

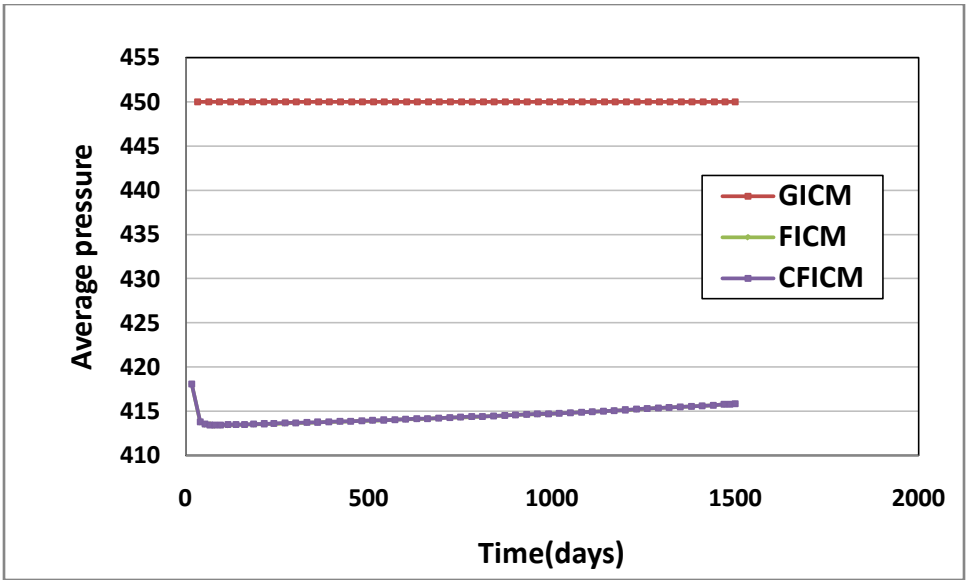

Figure 7.3.4-4: Average reservoir pressure curve vs. time.

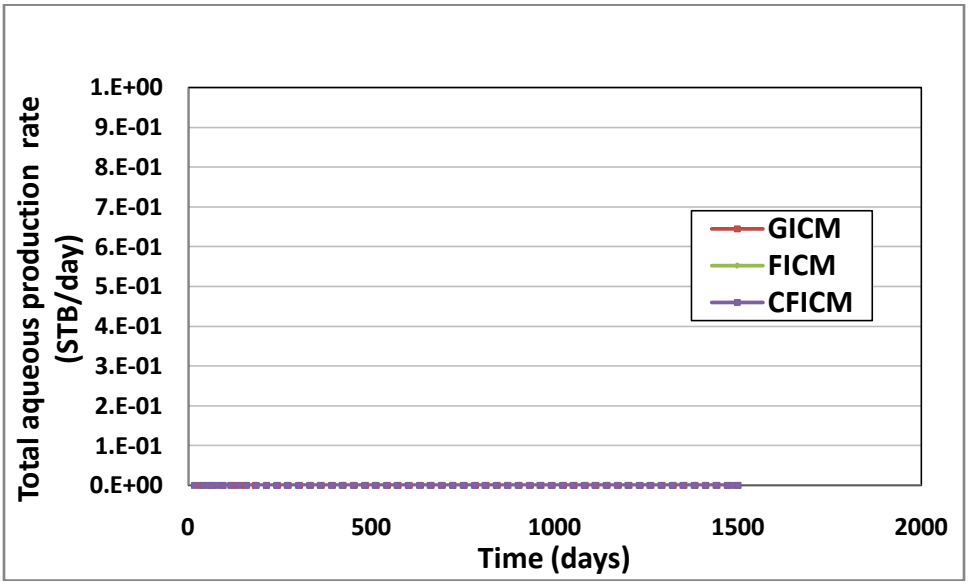

Figure 7.3.4-5: Total aqueous phase production (STB/day) vs. time.

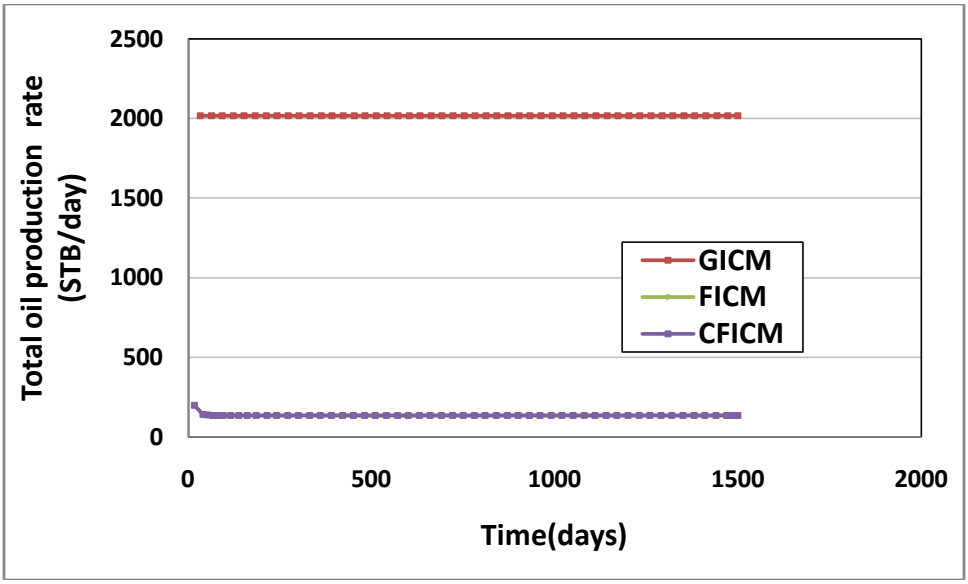

Figure 7.3.4-6: Total oil production (STB/day) vs. time.

7.3.4 Comments and Conclusions

Two iterative coupling methods have been compared to CFIM for four cases. FICM shows a good match with CFIM, and a similar efficiency of solving the four cases. GICM is shown to have the faster solving speed in most cases, but it is not a stable match with CFIM.

FICM was created partially based on PM. That is reason why FICM can match CFIM very well, and FICM can be faster than CFIM for some cases. GICM was created based on traditional iterative coupling methods. GICM convergence fully depends on the convergence of isotherm GPAS without any concern with the thermal module; it may obtain different results than CFIM, but it takes less time due to convergence properties.

Chapter 8: Summary, Conclusions, and Recommendations

8.1 Summary and Conclusions

Two new solvers, BoomerAMG, and SAMG, were implemented into three compositional simulators, i.e., UTCHEM, UTCOMP, and GPAS. A partition method and three iterative coupling methods were then performed and tested using GPAS. The following is summary of all research tasks that have been finished in this thesis:

1. SAMG has been installed and activated on the "petros" server at the Department of Petroleum and Geosystems Engineering at UT-Austin with the help of the SAMG Group, and especially with the great help of both Professor Klaus Stüben and Ms. Resi. SAMG was tested using the tests in SAMG's package.
2. HYPRE Version 2.40 was installed and activated on the "petros" server at the Department of Petroleum and Geosystems Engineering at UT-Austin. HYPRE was tested using the examples in HYPRE package.
3. PETSc was installed and activated on the "petros" server at the Department of Petroleum and Geosystems Engineering at UT-Austin. PETSc has been tested with PETSc package.
4. BoomerAMG and SAMG solvers have been implemented and coupled to UTCOMP as options. The implementation was carried out based on the rule: less modification of the original UTCOMP and the high efficiency of both SAMG and BoomerAMG for solving UTCOMP's linear system equations. Unfortunately, BoomerAMG cannot solve the linear system equations in UTCOMP because the linear systems are not symmetrical. Three cases such as solvent injection test, CO_2 injection test and water injection test were chosen to verify the efficiency of SAMG against UTCOMP's internal solvers, i.e., BCG32, BCG33, and BCG34. The numerical results show that

SAMG is the fastest solver used in UTCOMP for simulating the CO_2 injection and water injection processes, but slower than BCG34 for the solvent injection process. SAMG can solve larger and more accurate non-symmetrical linear system equations faster than the internal solvers of UTCOMP.

5. SAMG and BoomerAMG were implemented into UTCHEM. Since all linear system equations for the pressure equation from UTCHEM are SPD, both BoomerAMG and SAMG can solve the linear systems. Two kinds of tests were used for comparing the efficiency of three solvers SAMG, BoomerAMG, and JCG in UTCHEM consisting of both the homogeneous and the heterogeneous permeability test cases. The homogeneous-permeability test cases included Case I: water injection case. The heterogeneous tests included Case II: SP flooding case with constant permeability in the Y-direction and the Z-direction, and the heterogeneous permeability in the X-direction but keeping a constant permeability in each layer on a X-Y plane, Case III: SP flooding with constant permeability in the Z-direction, and heterogeneous permeabilities in the X-direction and the Y-direction but keeping the constant permeability in each layer on a X-Y plane, Case IV: SP flooding with heterogeneous permeabilities in the X-direction that generated with DPC at DPCS=0.5, and a constant permeability on a layer on the X-Y plane for the Y-direction and the Z-direction, Case V: SP flooding with permeabilities in the X-direction, the Y-direction, and the Z-direction were generated with DPC at DPCS=0.5, DPCS=0.7 and DPCS=0.9 respectively. The numerical results show that SAMG is the fastest solver among the three solvers; BoomerAMG is the second, and JCG is the slowest solver. SAMG performed better than the other solvers when solving cases with heterogeneous permeabilities. BoomerAMG had a similar trend to

SAMG, but much slower than SAMG. JCG is much slower than BoomerAMG for the given case study.

6. SAMG was implemented into GPAS Version 3.5 for both EOS compositional model and chemical flooding model based on rules of compatibility and extensibility and less modifications. A new index “LSOLVER” that can select either PETSc or SAMG has been added into the GPAS version 3.5. Four tests including Case I: gas injection with a one-component fluid reservoir, Case II: water injection with a one-component fluid reservoir, Case III water injection with a two-component fluid reservoir, and Case IV: water injection with a three-component fluid reservoir, were designed to verify the efficiency of SAMG against PETSc. The numerical results show that SAMG is at least 52% faster than PETSc solver, and SAMG will become much faster than PETSc when the number of components of the reservoir fluid increases for the given tests.
7. A PM between a thermal module and an isotherm GPAS was implemented into GPAS. The optimal preconditioner for both the Schur complement and the isothermal GPAS was built in the solving procedure of a partition method. Three tests include the different number of components and a different number of gridblocks. The numerical results show that, with an increase of number of gridblocks and components, PM became much faster than CFIM while maintaining very good match with CFIM results.
8. Two feasible iterative coupling methods, i.e., FICM and GICM, have been implemented into GPAS with a thermal module. The index of selecting iterative coupling methods employed the same index “LSOLVER” as that of linear solver. The implementation of iterative coupling involved the convergence control of nonlinear Newton iteration in GPAS. Four tests including a two-component water flooding

with10x10x5 gridblocks, a two-component water flooding with 20x20x5 gridblocks, a three-component water flooding with 12x12x5 gridblocks, and a four-component water flooding with 18x18x3 gridblocks were used for verifying the efficiency of iterative coupling methods against CFICM. For the same given tests, FICM can be faster than CFIM while keeping a good match with CFIM results; GICM is faster than CFIM but keeps less accuracy in matching with CFIM than FICM for the results.

8.2 Recommendations to Future Work

The following are the recommendations for future work.

1. Although the implementation of BoomerAMG and SAMG was carried out based on the rule: less modification of original UTCOMP and the high efficiency of both SAMG and BoomerAMG for solving UTCOMP's linear system equations, there are improvements that can be made in the implementation in the future as follows:
   a. UTCOMP can be modularized in a better way. UTCOMP needs a friendly interface to separate and connect solver part with main module parts, so that implementation of a new solver technique can be easier and safer. The solver part can be an independent module of UTCOMP instead of mixed and non-separated code.
   b. There were three tests for verifying the efficiency of SAMG against UTCOMP's internal solvers, but more tests should be performed if we want to know the reliable efficiency of SAMG. The tests can be divided into different injection fluids and different components of reservoir fluids, and number of gridblocks of reservoir. Then the efficiency of SAMG with different injection fluids, number of component of reservoir fluid, and number gridblocks can be seen clearly for these future tests.

c. Optimal parameters of SAMG for solving UTCOMP's linear system of equations are hard to determine. SAMG has so many flexible parameters to choose for solving a set of linear systems that the "optimal parameters" in this thesis may not have used the best optimal parameter for SAMG. More than likely, a better selection of parameters for SAMG is possible.

d. BoomerAMG cannot be applied for solving the UTCOMP's linear systems. Further improvement of BoomerAMG can be possible in the future. Once BoomerAMG can solve UTCOMP's linear systems, the option of BoomerAMG can be used in the future.

2. SAMG and BoomerAMG were implemented into UTCHEM, but the following issues may be improved in the future.

a. Further modularization of UTCHEM. UTCHEM is a modular simulator written in FORTRAN 90, but the linear solver part is not modularized very well in UTCHEM. The friendly interface between main modules and linear solver module can be built separately in the future. Thus, the complexions of implementing a new solver can be much easier and safer.

b. More variety of tests with different classifications such as different chemical injections, a different number of components, and a different number of gridblocks of reservoirs are needed for the complete comparison for SAMG, BoomerAMG and JCG. More comparisons may show more reliable information of the differences of efficiencies of three solvers. More tests may take very a long time, but it would be necessary for real oilfield applications.

c. Theoretical analysis may be carried out in the future. Since the properties of pressure linear systems depend on the chemical injection, components of reservoir fluids, and production of condition, these conditions can be very helpful for

analyzing the property of linear systems for the pressure equation. Then, how BoomerAMG and SAMG take the advantages of those properties to solve the systems efficiently can be more interesting research topic in the future. The results of theoretical analysis would be very helpful for selecting the SAMG's optimal parameters easily.

d. The parameters of SAMG used in this thesis may not be best even though the effort to optimize them have been made for the given tests. Many parameters and many choices for each parameter make a huge amount of possibilities for a given test. It is seldom possible to exhaust them completely for a group of optimal parameters in a limited period time of this thesis. Any user may choose a better group of SAMG's parameters if he/she wants to try more choices.

e. BoomerAMG is much slower than SAMG for our cases, but BoomerAMG may be improved in its efficiency if HYPRE's group can supply more useful inner development of information of BoomerAMG as SAMG group did for this study. Since BoomerAMG is free, the increasing BoomerAMG's speed would be an important issue in the future.

3. The further possible improvements for implementation of SAMG into GPAS Version 3.5 are started in the following.

   a. Modularization of linear solver in GPAS. GPAS version 3.5 has only the linear solver PETSc. For the linear solver of PETSc in GPAS, it just needs a coefficient matrix and a right hand vector for a Newtonian iteration, but the current implementation of PETSc has three steps mixed together in one subroutine: one is taking coefficient matrix one block line by one block line, another is setting up PETSc solver, finally solving the linear systems. It was very hard to do the implementation of SAMG in GPAS code because the separation between

transferring data into SAMG data structure in a subroutine has to be done first, and then the data into SAMG solver in another subroutine need to be transferred next. If the implementation of PETSc was implemented in this way, that would be more convenient for a new developer to add a new solver without spending too much time on inferring the data structure of PESTc's solver from the GPAS' source code only.

b. More tests with more specified classifications should be performed for future application. There are three kinds of classification such as injection types, number of components of reservoir fluids, and number of gridblocks. If the time is allowed, the more tests would be better for drawing more reliable conclusions for efficiency of SAMG in GPAS.

c. SAMG is a grid-based AMG solving procedure for GPAS. In the future, if possible, applying more SAMG theoretical analysis to our tests would be helpful to further improve the efficiency of SAMG.

d. The parallel of implementation of SAMG cannot work yet due to the lack of the complete documents of the parallel data structure of PETSc for GPAS. Domain decomposition of data structure of PETSc could not be figured out only based on the source code with GPAS in this thesis. In the future, if possible, further discussion with PETSc group would be helpful to figure out the index set division for each domain. The parallel SAMG should be working with very little modification to the current implementation of SAMG.

4. Possible improvements on the partition method between a thermal module and an isotherm GPAS that has been implemented into GPAS are the following:

   a. A parallel version of partition method can be done in the future. The reason why the parallel version of partition method didn't work in this thesis is due to a lack

of assistance from the PETSc developer for GPAS even I did my best to contact with PETSc group, and there is not any document available for the PETSc solver in GPAS. With the help of the developers of PETSc, the parallelization of partition method should be no difficulty based on what author has done in this thesis.

b. The parallelization of preconditioner of the Schur complement can be done in the future. It was difficult to construct the preconditioner of the Schur complement for the partition method, so the construction of parallel preconditioner of Schur complement could be more complicated. However, the parallel preconditioner of a Schur complement should be feasible if given enough time.

c. Better preconditioner strategy may be applied for the partition method. Other approaches may be possible for improving the efficiency of the partition method in the future, for example, constructing precoditioner based on the Schur complement instead of A_{22} would be possible in the future, and that would be much better than the currently used preconditioner.

5. Possible future work on three iterative coupling methods is presented in the following:

   a. Improvement of GICM. GICM is fast, but it is not very stable compared to CFIM. The improvement for the stability of GICM should be investigated.

   b. LICM's convergence is another challenge. With less flexibility, LICM can possibly converge if some suitable conditions can be added to LICM. Imposing conditions to guarantee the LICM's convergence and stability is important for future work with LICM.

Nomenclatures

C_p: Heat capacity

d: Diffusion length

D: Depth

f: Fugacity

h: Molar enthalpy

H: Enthalpy

PI: Productivity index

k_r: Relative permeability

K: Equilibrium ratio

$\vec{\vec{K}}$: Permeability tensor

n_c: Number of hydrocarbon components

N: Number of moles

P: Pressure

$\dot{q}$: Fluid production/injection rate

$\dot{Q}_L$: Heat-loss

R: Reaction rate

t: Time

T: Temperature

U: Internal energy

$\vec{v}$: Volumetric flux

V: Bulk volume

x: Mole fraction

Z: Compressibility factor

δ: Binary interaction coefficient

ϕ: Porosity

φ: Fugacity coefficient

γ: Specific gravity

η: Thermal diffusivity

λ_r: Relative mobility

μ: Viscosity

ω: Acentric factor

ζ: Phase molar density

Subscript

a: Aqueous

b: Bulk

g: Gas phase

i: Component index

j: Phase index

Phase index

o: Oleic

oil: Oil phase

r: Relative

v: Vapor

w: Well

Superscript

T: Total

Acronym

FDM: Finite Difference Method

BiCG: Biconjugate Gradient

CGS: Conjugate Gradient Squared

CG: Conjugate Gradient

BiCG: Biconjugate Gradient

JCG: CG iterative method plus Jacobi preconiditioner

CFIM: Classical Fully Implicit Method

SICM: Settari's iterative coupling method

TICM: Tran et al.'s iterative coupling method

CIFCM: Chin's iteratively, fully coupled method

SICM: Samier's classic iterative coupling method

SAGD: Steam-assisted gravity drainage

GICM: General, iterative coupling method

FICM: Fully, iterative coupling method

LICM: Loose, iterative coupling method

SPICM: Samier's practical iterative coupling method

WIP: Water in place

HCIP: Hydrocarbon in place

OOIP: Origin oil in place

OGIP: Original gas in place

BOM: Black oil model

BHP: Bottomhole pressure

GOR: Gas oil ratio

WOR: Water oil ratio

HC: Hydrocarbon

MIP: Materials in place

CSR: Compress spars row

SPD: Symmetric, positive and definite

DPC: Dykstra-Parsons coefficient

Appendix A: User's Manual of UTCOMP with SAMG

This section presents the user's guide of the new version of UTCOMP with a new solver SAMG.

A.1 Keywords

All keywords are listed in the following table

IPRESS	Solver
3	BCG
4	BCGS
5	ORTNORMIN
6	SAMG

IPREC	Preconditioner
2	Jacobi
3	ILU
4	Modified ILU

A.2 INPUT File

```
CC*********************************************************************
CC                                                                    *
CC                                                                    *
CC                                                                    *
CC*********************************************************************
CC                                                                    *
CC                                                                    *
CC                                                                    *
CC LENGTH(FT):          INJECTION FLUID  : none                       *
CC HEIGHT(FT):          INJECTION RATE   : none                       *
CC WIDTH (FT):          W/O REL. PERM    : none                       *
CC POROSITY  :          G/O REL. PERM    : none                       *
CC ABS. PERM :          3-PHASE REL. PERM: none                       *
CC TEMP(F)   :        WETTIBILITY:       none                         *
CC PRE(PSI)  :        W/O CAP. PRESSURE: none                        *
CC SOR       :       G/O CAP. PRESSURE: none                         *
CC SWC       :       DISPLACEMENT TYPE:                              *
CC Grid Dim. :          Trapping Model   : no                         *
CC                                                                    *
```

```
CC                                                                      *
CC                                                                      *
CC                                                                      *
CC                                                                      *
CC                                                                      *
CC**********************************************************************
CC                                                                       *
CC                                                                       *
CC                                                                       *
CC**********************************************************************
CC
CC.+..1..+..2..+..3..+..4..+..5..+..6..+..7.
CC CASE NAME WITH FORMAT ( 17A4, A2 ) OF TOTAL 70 COLUMNS.
*----HEADER
 6comp
CC
CC NUMBER OF COMPONENTS.
*--------NC
       6
CC COMPONENT NAMES WITH FORMAT ( 1X, A8 ), NC CARDS.
CC.+.8
*----NAME
  C1
  C3
  C6
  C10
  C15
  C20
CC
CC BLACK OIL OPTION; AQUIFER SALINITY (ppm); AQUIFER OPTION
*-----IBOST     SLNTY    IAQUIF
        0       0.        0
CC CRITICAL PRESS. (PSI), TEMP. (R) AND  VOL. (CU FT/LB-MOLE),
CC MOLECULAR WT. (LB/LB-MOLE), ACENTRIC FACTOR, PARACHOR. NC CARDS.
*--------PC       TC       VC       MW      OM    PARACH  VSP
      667.8    343.0    1.599   16.0     0.013   71     0.0
      616.3    665.7    3.211   44.1     0.152    151.0  0.0
      436.9    913.4    5.923   86.2     0.301    271.0  0.0
```

```
      304.0     1111.8    9.087  142.3    0.488    431.0   0.0
      200.0     1270.0    14.00696  206.0    0.650    631.0   0.0
      162.0     1380.0    19.484  282.0    0.850    831.0  0.0
CC EOS parameters (Ac and Bc)
CC NC CARDS.
*----PARAA              PARAB
  0.457235529       0.077796074
  0.457235529       0.077796074
  0.457235529       0.077796074
  0.457235529       0.077796074
  0.457235529       0.077796074
  0.457235529       0.077796074
CC
CC BINARY INTERACTION COEFFICIENTS, CIJ. NC CARDS.
*-----DELTA
0.00000
0.00000  0.00000
0.00000  0.00000 0.00000
0.00000  0.00000 0.00000 0.00000
0.00000  0.00000 0.00000 0.00000 0.00000
0.10000  0.00000 0.00000 0.00000 0.00000 0.00000
CC
CC BINARY INTERACTION COEFFICIENTS, DIJ. NC CARDS.
*-------DIJ
0.000
0.000    0.000
0.000    0.000 0.000
0.00000  0.00000 0.00000 0.00000
0.00000  0.00000 0.00000 0.00000 0.00000
0.0000   0.00000 0.00000 0.00000 0.00000 0.00000
CC
CC reduction method: (0: OFF, 1: ON)
*--------irfla  irsa
        0      0
CC
CC NC CARDS
*-------rh       rg
     1.00d0    1.0d0
```

```
    0.00d0    0.0d0
    0.00d0    0.01d0
    0.00d0    0.01d0
    0.00d0    0.01d0
    0.00d0    0.1d0
CC
CC MAXIMUM NUMBER OF PHASES ( 3 OR 4 )
*--------NP     IVISC     ISINGL   ISOLU
       3      1       1       0
CC IEOS: 1,    IPEM: 0 OR 1
CC ISTAM: -1, 0 OR 1, IEST: 0 OR 1  KI: 0, 1 OR 2
*---IEOS   IPEM     ISTAM   IEST  IVSP   KI
   1     1        -1     1     0     0
CC
CC ITERATION TOLERANCES FOR PERSCHKE'S FLASH ROUTINES.
*----TOLFLA    TOLFLM    TOLPD    TOLSAM    TOLSAS    TOLSUM
   1.0E-8   1.0E-8    1.0E-10  1.0E-08   1.0E-08   1.0E-08
CC
CC MAXIMUM NUMBER OF ITERATIONS FOR PERSCHKE'S FLASH ROUTINES.
*----MAXFLA    MAXFLM    MAXPD    MAXSAM    MAXSAS    MAXSANR
   3000      1000      1000     1000      3000      20
CC
CC tolline: tol for linesearch bisec: relaxation coefff
*-----icvxset   iwline   tolline   bisec
      1        1       1.d-3   0.9d0
CC
CC Linesearch using bracketing
*-----sigma   beta    maxm
    1.d-4   0.5d0   100
CC
CC VECTOR FLASH OPTION
*-----IVCFL    TOLVFL    MAXVFL
       1     1.E-10       30
CC
CC SWITCHING PARAMETERS FOR PERSCHKE'S FLASH ROUTINES.
*----SWIPCC    SWIPSA
    .001       0.001
CC
```

```
CC PHASE IDENTIFICATION PARAMETERS FOR PERSCHKE'S FLASH ROUTINES.
*------IOIL      ITRK   DMSLIM
       1       2      25.0
C
CC IFLAGT ( 0 : OFF,   1 : ON )
*------IFLAGT    IASPR
         0       0
CC
CC*************************************************************************
CC                                                                        *
CC OUTPUT OPTIONS                                                         *
CC                                                                        *
CC*************************************************************************
CC
CC
CC HISTORY PRINTING PARAMETER FOR <<HISTORY.CPR>>.
*---NHSSKIP   NSTSKIP     IPV
   250        250          0
CC
CC REFERENCE CONCENTRATION, CONC0, USED FOR EFFLUENT CONCENTRATION.
*-----CONC0
 1.0   1.0   1.0   1.0   1.0   1.0
CC
CC NUMBER OF PRINTS FOR <<.TAB>> (ALSO FOR TRAPPING & ASPHALTENE DATA)
*-------NPR
       1
CC
CC TIME(DAYS) AND FLAGS ( 0 OR 1 ) NPR CARDS.
*------TPR     MPRP    MPRSAT   MPROMFR   MPRPMFR    MPRPRO   MPRATES
     0.01     1       1        1         1          1        1
CC
CC NUMBER OF PRINTS FOR <<PROFILE.CPR>>.
*-------NPF
       1
CC
CC TIME(DAYS) AND FLAGS ( 0 OR 1 ) NPF CARDS.
*-------TPF    MPFSAT   MPFOMFR   MPFPMFR   MPFPROP
     0.01     1        1         1         1
```

```
CC
CC NUMBER OF PRINTS FOR <<CONTOUR.CPR>>.
*-------NCT
       1
CC
CC TIME(DAYS) AND FLAGS ( 0 OR 1 ) NCT CARDS.
*-------TCT     MCTP    MCTSAT   MCTOMFR   MCTPMFR    MCTPRO
      0.01     1       1        1         1          1
CC
CC**************************************************************************
CC                                                                         *
CC RESERVOIR AND WELL DATA                                                 *
CC                                                                         *
CC**************************************************************************
CC
CC A FLAG FOR RESERVOIR GEOMETRY:
CC 2-D: 11(Y), 12(X), 13(Z), 2-D: 21(XY), 22(YZ), 23(XZ), 3-D: 31
*-----IGEOM       INUG
      31         0
CC
CC NUMBER OF GRID BLOCKS IN X, Y, AND Z.
*--------NX        NY        NZ
         195       195       5
CC
CC NUMBER OF WELLS
*--------NW       IWM
        2         2
CC
CC WELLBORE RATIUS (FT). NW NUMBERS.
*--------RW: (NW)
     0.44      0.44
CC
CC WELL LOCATIONS. NW CARDS.
*-------LXW       LYW    IDIR   LZWF      LZWL
       1          1      3      1         5
      195       195      3      1         5
CC
CC A FLAG ( 0 OR 1 ) FOR GRID BLOCK SIZE IN X-DIRECTION.
```

```
*-------MDX
        0
CC
CC CONSTANT GRID BLOCK SIZE IN X-DIRECTION (FT).
*--------DX
       75.
CC
CC A FLAG ( 0 OR 1 ) FOR GRID BLOCK SIZE IN Y-DIRECTION.
*-------MDY
       0
CC
CC CONSTANT GRID BLOCK SIZE IN Y-DIRECTION (FT).
*--------DY
       75.
CC
CC A FLAG ( 0 OR 1 ) FOR GRID BLOCK SIZE IN Z-DIRECTION.
*-------MDZ
       0
CC
CC CONSTANT GRID BLOCK SIZE IN Z-DIRECTION (FT).
*--------DZ
       50.
CC
CC A FLAG ( 0 OR 1 ) FOR FORMATION DEPTH.
*--------MD
        0
CC
CC DEPTH (FT) OF THE MOST UPPER LAYER.
*---------D
      0.0
CC
CC A FLAG ( 0 OR 1 ) FOR FORMATION POROSITY.
*------MPOR
      0
CC
CC HOMOGENEOUS POROSITY (FRACTION) AT PF.
*----PORSTD
    0.3
```

```
CC
CC A FLAG ( 0 OR 1 ) FOR PERMEABILITY IN X-DIRECTION.
*----MPERMX
     0
CC
CC HOMOGENEOUS PERMEABILITY (MD) IN X-DIRECTION.
*-----PERMX
     100.
CC
CC A FLAG ( 0 OR 1 ) FOR PERMEABILITY IN Y-DIRECTION.
*----MPERMY
     0
CC
CC HOMOGENEOUS PERMEABILITY (MD) IN Y-DIRECTION.
*-----PERMY
     100.
CC
CC FLAG ( 0 OR 1 ) FOR PERMEABILITY IN Z-DIRECTION.
*----MPERMZ
     0
CC
CC HOMOGENEOUS PERMEABILITY (MD) IN Z-DIRECTION.
*-----PERMZ
    100.
CC
CC FORMATION COMPRESSIBILITY (1/PSI) AND REFERENCE PRESSURE (PSI).
*--------CF          PF
     50.0e-6       3100
CC H2O COMPRESSIBILITY (1/PSI), REFERENCE PRESSURE (PSI) AND
CC MOLAR DENSITY (LB-MOLE/CU FT).
*--------CW       PW     DENMWS
   3.0e-6     14.65      3.467
CC
CC WATER MOLECULAR WT. (LBM/LBM-MOLE) AND VISCOSITY (CP).
*-------WTW     VISCW
      18.      1.0
CC
CC FORMATION TEMPERATURE (F).
```

```
*-----TEMPF
     150.0
CC
CC STANDARD TEMPERATURE (F) AND STANDARD PRESSURE (PSI).
*-----TFSTD     PSTD
      60.     14.65
CC
CC A FLAG ( 1, 2, 3 OR 4 ) FOR NUMERICAL DISPERSION CONTROL.
*----IUPSTW
       1
CC
CC ITC ( 0 : NO 2ND ORDER TIME,   1 : 2ND ORDER TIME ON )
*----ITC
     0
CC RESTART OPTIONS.
CC ISTART ( 1 OR 2 ), ISTORE ( 0 OR 1 ).
*----ISTART    ISTORE
       1        0
CC
CC A FLAG ( 0 OR 1 ) FOR AUTOMATIC TIME-STEP SELECTION ( = 1 ).
*-------MDT
       1
CC
CC A FLAG ( 0 OR 1 ) FOR PHYSICAL DISPERSION CALCULATION.
*-----MDISP
      0
CC FLAGS FOR RELATIVE PERMEABILITY MODEL AND CAPILLARY PRESSURE.
CC IPERM ( 1 OR 2 ), ICPRES ( 0 OR 1 ).
*-----IPERM    ICPRES     ICAP    IRPERM
     2       0         0         0
CC CAPILLARY PRESSURE PARAMETERS AND
CC WATER/OIL INTERFACIAL TENSION (DYNES/CM).
*-------EPC      CPC    RIFTWO    RIFTWG    RIFTWL
       2.       2.0    20.       24.       30.
CC
CC HIGH IFT RESIDUAL SATURATIONS.
*------S1RW     S2RW1     S2RW2     S3RW     S4RW1     S4RW2
      0.25     .000001      .000001     0.0      0.0     0.0
```

```
CC
CC LOW IFT RESIDUAL SATURATIONS.
*------S1RC     S2RC1     S2RC2     S3RC     S4RC1     S4RC2
     0.0    .0      .0        0.0        0.0        0.0
CC
CC HIGH IFT END POINT RELATIVE PERMEABILITY.
*------P1RW      P2RW      P3RW      P4RW
      0.3       1.0       1.0       1.0
CC
CC LOW IFT END POINT RELATIVE PERMEABILITY.
*------P1RC          P2RC        P3RC     P4RC
      1.0          1.0         1.0      1.0
CC
CC HIGH IFT EXPONENT OF RELATIVE PERMEABILITY.
*-------E1W      E2W1      E2W2       E3W      E4W1      E4W2
      1.       1.0       1.0       1.0      1.0       1.0
CC
CC LOW IFT EXPONENT OF RELATIVE PERMEABILITY.
*-------E1C      E2C1      E2C2       E3C      E4C1      E4C2
      1.0       1.0       1.0       1.0       1.0       1.0
CC
CC WATER AND L1 PHASE CAPILLARY DESATURATION PARAMETERS.
*-------T11       T12      T211      T221      T212      T222
       0         0        0         0         0         0
CC
CC GAS AND L2 PHASE CAPILLARY DESATURATION PARAMETERS.
*-------T31       T32      T411      T421      T412      T422
       0         0        0         0         0         0
C
CC A FLAG FOR PRESSURE EQUATION SOLVER ( 1, 2, 3, 4 OR 5 ).
*----IPRESS    IPREC   METHSL   OMEGA
        6      2       1        1.0
CC
CC ITERATIVE PRESSURE SOLVER PARAMETERS.
*-----ITMAX  LEVLIT  IDGTS   NS1       NS2      ZETA
    100000     1      1      5      1000000   1.E-10
CC
CC INITIAL TIME (DAYS).
```

```
*---------T
        0
CC
CC A FLAG ( 0 OR 1 ) FOR INITIAL PRESSURE.
*--------MP
       0
CC
CC CONSTANT INITIAL PRESSURE (PSIA).
*---------P
            3100.0
CC
CC A FLAG ( 0 OR 1 ) FOR INITIAL WATER SATURATION.
*------MSAT
      0
CC
CC CONSTANT INITIAL WATER SATURATION (FRACTION).
*-------SAT
            0.25
CC
CC A FLAG ( 0 OR 1 ) FOR INITIAL OVERALL COMPOSITION.
*-----MOMFR
      0
CC
CC CONSTANT INITIAL COMPOSITION (MOLE FRACTION).
*------OMFR
0.5
0.03
0.07
0.2
0.15
0.05
CC
CC*************************************************************************
CC                                                                        *
CC RECURRENT DATA                                                         *
CC                                                                        *
CC*************************************************************************
CC
```

```
CC
CC MAXIMUM TIME (DAYS), TIME STEP (DAYS) AND WELL DATA.
*----TM          DT       NWELLS    GORLIM    WORLIM
  300.0         0.0002      2         -1.       -1.
CC
CC PARAMETERS FOR TIME STEP SELECTORS.
*-----DTMAX    DTMIN    DSLIM    DPLIM     DVLIM    DMFACT
     5        1e-4    0.05      0.01      .05     .05
CC
CC WELL NO. AND WELL TYPE.
*--------LW    IQTYPE
       1     4
CC
CC   (STB/D)  (MSCF/D)
*----QPSVC(1)   QPSVC(3)   NCOMP   ISWITCH   PBHC
     5000.      0        1       0       3100
CC
cc overall components
*---------KC   Z1
       7     1.
CC
CC WELL NO. AND WELL TYPE.
*--------LW    IQTYPE
       2      -2
CC
CC CONSTANT BHP PRODUCER
*--------PBHC
      3100.0
CC
CC END OF INPUT.
*--------TM DT    NWELLS    GORLIM    WORLIM ----------------
             -1.  -1.  -1      -1.E10    -1.E10
```

Appendix B: User's Manual of UTCHEM with BoomerAMG and SAMG

This appendix will provide the user's guide for the new version of UTCHEM with new solvers, HYPRE and SAMG.

B.1. Keywords

A new input file named as "solver.itck" was added with original input files "INPUT" and "HEAD."

B.2 New Input File "solver.itck"

```
1 JCG 2 SAMG 3 BoomerAMG
2
```

B.3 Input File "HEAD"

```
water1
NX    NY    NZ    N    NWELL
64    64    8     8    2
NTW       NTA
0         0
NO        NPHAS
0         3
NSUB      MSUB
0         0
```

B.4 Input File “INPUT”

```
CC*************************************************************************
CC                                                                       *
CC    BRIEF DESCRIPTION OF DATA SET : UTCHEM (VERSION 9.9 )              *
CC                                                                       *
CC*************************************************************************
CC                                                                       *
CC  SURFACTANT/POLYMER FLOOD TEST (EX01), 11X11X2                        *
CC                                                                       *
CC  LENGTH (FT) : 250            PROCESS : waterflood                    *
CC  THICKNESS (FT) : 10          INJ. RATE (FT3/DAY) : 112.3             *
CC  WIDTH (FT) : 250             COORDINATES : CARTESIAN                 *
CC  POROSITY : 0.20                                                      *
CC  GRID BLOCKS : 64X64X8                                                *
CC  DATE : August 2010                                                   *
CC                                                                      *
CC*************************************************************************
CC
CC*************************************************************************
CC                                                                       *
CC   RESERVOIR DESCRIPTION                                               *
CC                                                                       *
CC*************************************************************************
CC
CC
*----RUNNO
water1
CC
CC
*----HEADER
EX01-icwi=-1, run with GUI
TESTING  UTCHEM VERSION 9.9
water flood
CC
CC SIMULATION FLAGS
*---- IMODE IMES IDISPC ICWM ICAP IREACT IBIO ICOORD ITREAC ITC IGAS IENG
      1   2   3     0   0    0     0   1    0     0   0  0
```

```
CC
CC NO. OF GRIDBLOCKS,FLAG SPECIFIES CONSTANT OR VARIABLE GRID SIZE, UNIT
*----NX   NY     NZ  IDXYZ   IUNIT
   64   64     8   0       0
CC
CC  CONSTANT GRID BLOCK SIZE IN X, Y, AND Z
*----DX1       DY1      DZ1
   20        20       5.0
CC
CC TOTAL NO. OF COMPONENTS, NO. OF TRACERS, NO. OF GEL COMPONENTS
*----N   NO  NTW NTA  NGC NG  NOTH
   6   0 0   0   0    0    0
CC
CC  NAME OF SPECIES
*---- SPNAME(I) FOR I=1,N
WATER
OIL
surf
polymer
anion
calcium
CC
CC FLAG INDICATING IF THE COMPONENT IS INCLUDED IN CALCULATIONS OR NOT
*----ICF(KC) FOR KC=1,N
   1  1  1  1  1  1
CC
CC**********************************************************************
CC                                                                     *
CC    OUTPUT OPTIONS                                                   *
CC                                                                     *
CC**********************************************************************
CC
CC
CC FLAG FOR PV OR DAYS TO PRINT OR TO STOP THE RUN
*----ICUMTM  ISTOP  IOUTGMS
     0        0    0
CC
CC FLAG INDICATING IF THE PROFILE OF KCTH COMPONENT SHOULD BE WRITTEN
```

```
*----IPRFLG(KC),KC=1,N
   0 0  0 0 0 0
CC
CC FLAG FOR individual map files
*----IPPRES IPSAT IPCTOT IPBIO IPCAP IPGEL IPALK IPTEMP IPOBS
    0     0     0     0     0    0  0    0      0
CC
CC FLAG for individual output map files
*----ICKL IVIS IPER ICNM ICSE ihystp  ifoamp  inoneq
    0   1   0  0   0 0       0      0
CC
CC FLAG  for variables to PROF output file
*----Iads IVel Irkf Iphse
    0   0   0   0
CC
CC*********************************************************************
CC                                                                    *
CC   RESERVOIR PROPERTIES                                             *
CC                                                                    *
CC*********************************************************************
CC
CC
CC MAX. SIMULATION TIME ( DAYS)
*---- TMAX
     500
CC
CC ROCK COMPRESSIBILITY (1/PSI), STAND. PRESSURE(PSIA)
*----COMPR   PSTAND
    0.     0.
CC
CC FLAGS INDICATING CONSTANT OR VARIABLE POROSITY, X,Y,AND Z PERMEABILITY
*----IPOR1 IPERMX IPERMY IPERMZ  IMOD   ITRNZ    INTG
     0     0    1     0     0     0       0
CC
CC CONSTANT POROSITY
*----PORC1
    .20
CC
```

```
CC CONSTANT X-PERMEABILITY (MILIDARCY) FOR LAYER K = 1,NZ
*----PERMXC
      100
CC
CC CONSTANT Y-PERMEABILITY (MILIDARCY) FOR LAYER K = 1,NZ
*----PERMY(1)  PERMY(2)
   64*500.    64*100.  64*500.    64*100. 64*500.    64*100. 64*500.    64*100.
CC
CC CONSTANT Z-PERMEABILITY
*----PERMZC (MILIDARCY)
   50.
CC
CC FLAG FOR CONSTANT OR VARIABLE DEPTH, PRESSURE, WATER SATURATION
*----IDEPTH  IPRESS  ISWI  ICWI
    0       0       0   -1
CC
CC CONSTANT DEPTH (FT)
*----D111
    0.
CC
CC CONSTANT PRESSURE (PSIA)
*----PRESS1
   100
CC
CC CONSTANT INITIAL WATER SATURATION
*----SWI
   0.6
cc
cc
*--- c50    c60
  0.17    0.003
CC
CC*****************************************************************
CC                                                                 *
CC   PHYSICAL PROPERTY DATA                                        *
CC                                                                 *
CC*****************************************************************
CC
```

```
CC
CC OIL CONC. AT PLAIT POINT FOR TYPE II(+)AND TYPE II(-), CMC
*---- C2PLC   C2PRC  EPSME  IHAND
     0.     1.    .0001    0
CC
CC FLAG INDICATING TYPE OF PHASE BEHAVIOR PARAMETERS
*---- IFGHBN
       0
CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT SALINITY
CC FOR ALCOHOL 1
*----HBNS70 HBNC70 HBNS71 HBNC71 HBNS72 HBNC72
   0.131   .1    .191   .026   .363   .028
CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT SALINITY
CC FOR ALCOHOL 2
*----HBNS80 HBNC80 HBNS81 HBNC81 HBNS82 HBNC82
   0.    0.    0.    0.    0.    0.
CC
CC LOWER AND UPPER EFFECTIVE SALINITY FOR ALCOHOL 1 AND ALCOHOL 2
*----CSEL7  CSEU7  CSEL8  CSEU8
   .177   .344   0.    0.
CC
CC THE CSE SLOPE PARAMETER FOR CALCIUM AND ALCOHOL 1 AND ALCOHOL 2
*----BETA6  BETA7  BETA8
   .8    -2.    0.
CC
CC FLAG FOR ALCOHOL PART. MODEL AND PARTITION COEFFICIENTS
*----IALC  OPSK7O  OPSK7S  OPSK8O  OPSK8S
   1    0.     0.     0.     0.
CC
CC NO. OF ITERATIONS, AND TOLERANCE
*----NALMAX   EPSALC
   20      .0001
CC
CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=1
*----AKWC7   AKWS7  AKM7  AK7    PT7
   4.671   1.79   48.   35.31   .222
CC
CC ALCOHOL 2 PARTITIONING PARAMETERS IF IALC=1
```

```
*----AKWC8   AKWS8  AKM8  AK8    PT8
   0.     0.    0.   0.     0.
CC
CC
*---- IFT MODEL FLAG
    0
CC
CC INTERFACIAL TENSION PARAMETERS
*----G11  G12    G13   G21   G22    G23
   13.  -14.8   .007  13.   -14.5  .010
CC
CC LOG10 OF OIL/WATER INTERFACIAL TENSION
*----XIFTW
   1.3
CC
CC MASS TRANSFER FLAG
*----IMASS  icor
   0     0
CC
CC CAPILLARY DESATURATION PARAMETERS FOR PHASE 1, 2, AND 3
*----ITRAP   T11       T22       T33
   1      1865.     59074     364.2
CC
CC RELATIVE PERM. FLAG (0:IMBIBITION COREY,1:FIRST DRAINAGE COREY)
*----IPERM  IRTYPE
   0      0
CC
CC FLAG FOR CONSTANT OR VARIABLE REL. PERM. PARAMETERS
*----ISRW  IPRW  IEW
   0     0   0
CC
CC CONSTANT RES. SATURATION OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*----S1RWC  S2RWC  S3RWC
   .37   .35   .37
CC
CC CONSTANT ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*----P1RW  P2RW  P3RW
   .11   .95   .11
```

```
CC
CC CONSTANT REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*----E1W  E2W  E3W
   1.  2.16  1.
CC
CC CONSTANT RES. SATURATION OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*----S1RWC  S2RWC  S3RWC
   0   .0   .0
CC
CC CONSTANT ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*----P1RW  P2RW  P3RW
   1   1   1
CC
CC CONSTANT REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*----E1W  E2W  E3W
   1.  2.16  1.
CC
CC WATER AND OIL VISCOSITY , RESERVOIR TEMPERATURE
*----VIS1   VIS2  TEMPV
   0.86  4.  0.
CC
CC VISCOSITY PARAMETERS
*----ALPHA1 ALPHA2  ALPHA3  ALPHA4  ALPHA5
   2.5    2.3    10.   1.    1.
CC
CC PARAMETERS TO CALCULATE POLYMER VISCOSITY AT ZERO SHEAR RATE
*----AP1    AP2    AP3
   81.    2700.  2500.
CC
CC PARAMETER TO COMPUTE CSEP,MIN. CSEP, AND SLOPE OF LOG VIS. VS. LOG CSEP
*----BETAP CSE1  SSLOPE
   10.   .01  .17
CC
CC PARAMETER FOR SHEAR RATE DEPENDENCE OF POLYMER VISCOSITY
*----GAMMAC  GAMHF  POWN      IPMOD
   20.    10.   1.8       0
CC
CC FLAG FOR POLYMER PARTITIONING, PERM. REDUCTION PARAMETERS
```

```
*----IPOLYM EPHI3 EPHI4 BRK    CRK      RKCUT
   1     1.   0.8    1000.  0.0186   10
CC
CC SPECIFIC WEIGHT FOR COMPONENTS 1,2,3,7,AND 8 , AND GRAVITY FLAG
*----DEN1  DEN2   DEN23  DEN3 DEN7 DEN8 IDEN
   .433  .368   0.368  .42  .346  0.  2
CC
CC  FLAG FOR CHOICE OF UNITS ( 0:BOTTOMHOLE CONDITION , 1: STOCK TANK)
*-----ISTB
    0
CC
CC COMPRESSIBILITY FOR VOL. OCCUPYING COMPONENTS 1,2,3,7,AND 8
*----COMPC(1)  COMPC(2)  COMPC(3)  COMPC(7)  COMPC(8)
   0.       0.       0.       0.       0.
CC
CC CONSTANT OR VARIABLE PC PARAM., WATER-WET OR OIL-WET PC CURVE FLAG
*----ICPC   IEPC  IOW
   0       0     0
CC
CC CAPILLARY PRESSURE PARAMETER, CPC0
*----CPC0
  0.
CC
CC CAPILLARY PRESSURE PARAMETER, EPC0
*---- EPC0
    2.
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 1 (D(KC),KC=1,N)
*----D(1) D(2) D(3) D(4) D(5) D(6) D(7) D(8) D(9) D(10) D(11)
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.    0.
CC
CC MOLECULAR DIFFUSION COEF. OF KCTH COMPONENT IN PHASE 2 (D(KC),KC=1,N)
*----D(1) D(2) D(3) D(4) D(5) D(6) D(7) D(8) D(9)  D(10)  D(11)
   0.  0.  0.  0.  0.  0.  0.  0.  0.    0.     0.
CC
CC MOLECULAR DIFFUSION COEF. OF KCTH COMPONENT IN PHASE 3 (D(KC),KC=1,N)
*----D(1) D(2) D(3) D(4) D(5) D(6) D(7) D(8) D(9) D(10)  D(11)
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.    0.
```

```
CC
CC LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 1
*----ALPHAL(1)    ALPHAT(1)
   12.         .4
CC
CC LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 2
*----ALPHAL(2)    ALPHAT(2)
   12.         .4
CC
CC LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 3
*----ALPHAL(3)    ALPHAT(3)
   12.         .4
CC
CC FLAG TO SPECIFY ORGANIC ADSORPTION CALCULATION
*----IADSO
    0
CC
CC SURFACTANT AND POLYMER ADSORPTION PARAMETERS
*----AD31  AD32  B3D    AD41   AD42  B4D   IADK, IADS1, FADS REFK,md
   1.5    .5    1000.  0.7    0.    100.  0     0     0    500
CC
CC PARAMETERS FOR CATION EXCHANGE OF CLAY AND SURFACTANT
*----QV     XKC   XKS  EQW
   0.0   0     0   419.
CC
CC*********************************************************************
CC                                                                    *
CC    WELL DATA                                                       *
CC                                                                    *
CC*********************************************************************
CC
CC
CC FLAG FOR RIGHT AND LEFT BOUNDARY
*---- IBOUND izone
    0      0
CC
CC TOTAL NUMBER OF WELLS, WELL RADIUS FLAG, FLAG FOR TIME OR COURANT NO.
*----NWELL   IRO   ITSTEP  NWREL
```

```
    2     2     1     2
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*----IDW   IW    JW    IFLAG    RW     SWELL  IDIR   IFIRST  ILAST  IPRF
    1   1     1     1      .5    0.     3     1       8     0
CC
CC WELL NAME
*---- WELNAM
INJECTOR
CC
CC ICHEK, MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*----ICHEK PWFMIN   PWFMAX  QTMIN   QTMAX
    0    0.0     5000.   0.0     1000.
CC
CC WELL ID, LOCATION, AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*----IDW  IW   JW   IFLAG    RW     SWELL  IDIR  IFIRST     ILAST    IPRF
   2    64   64    2      .5      0.     3     1       8      0
CC
CC WELL NAME
*---- WELNAM
PRODUCER
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*----ICHEK PWFMIN   PWFMAX  QTMIN   QTMAX
    0    0.0     5000.   0.0     50000.
CC
CC ID,INJ. RATE AND INJ. COMP. FOR RATE CONS. WELLS FOR EACH PHASE (L=1,3)
*----ID  QI(M,L)  C(M,KC,L)
   1    4000       0.97    0.  0.03  0.15  0.17  0.001  0.0  0.  1. 1.  1.
   1    0.         0.  0.  0.    0.   0.   0.     0.    0.  0. 0.  0.
   1    0.         0.  0.  0.    0.   0.   0.     0.    0.  0. 0.  0.
CC
CC ID, BOTTOM HOLE PRESSURE FOR PRESSURE CONSTRAINT WELL (IFLAG=2 OR 3)
*----ID   PWF
   2    100
CC
CC CUM. INJ. TIME , AND INTERVALS (PV OR DAY) FOR WRITING TO OUTPUT FILES
*----TINJ    CUMPR1   cumh1  WRHPV   WRPRF      RSTC
```

```
   200   50       50     10     5      0.25
CC
CC FOR IMES=2 ,THE INI. TIME STEP,CONC. TOLERANCE,MAX.,MIN. COURANT NUMBERS
*----DT      DCLIM   CNMAX   CNMIN
   0.05    0.03     0.2      0.01
cc
cc
*--- ibmod
     0
CC
CC IRO, ITIME, NEW FLAGS FOR ALL THE WELLS
*---- IRO ITSTEP IFLAG
      2   1     1  2
CC
CC NUMBER OF WELLS CHANGES IN LOCATION OR SKIN OR PWF
*----NWEL1
     0
CC
CC NUMBER OF WELLS WITH RATE CHANGES, ID
*----NWEL1   ID
   1        1
CC
CC ID,INJ. RATE AND INJ. COMP. FOR RATE CONS. WELLS FOR EACH PHASE (L=1,3)
*----ID  QI(M,L)  C(M,KC,L)
   1    4000     1.  0.  0.  0.15  0.17  0.001  0.  0.  0. 0. 0.
   1    0.       0.  0.  0.  0.    0.   0.     0.  0.  0. 0. 0.
   1    0.       0.  0.  0.  0.    0.   0.     0.  0.  0. 0. 0.
CC
CC CUM. INJ. TIME , AND INTERVALS (PV) FOR WRITING TO OUTPUT FILES
*----TINJ    CUMPR1 cumh   WRHPV     WRPRF   RSTC
   3300    360     360     30       360    730
CC
CC FOR IMES=2 ,THE INI. TIME STEP,CONC. TOLERANCE,MAX.,MIN. TIME STEPS
*----DT      DCLIM   CNMAX   CNMIN
   0.05    0.03     0.2    0.02
```

Appendix C: User's Manual of GPAS with SAMG

This section will briefly present a user's guide of the new version of GPAS with SAMG solver.

C.1 New Added Index

LSOLVER	SOLVER's name
0	PETSc
1	SAMG

C.2 New Added Index

```
$LSOLVER:0 PETSc,1: SAMG
LSOLVER = 1
```

C.3 New Input File

```
$LSOLVER:0 petsc,1: samg
LSOLVER = 1

TITLE(2)="3-D SIX COMPONENT GAS INJECTION"

DESCRIPTION()=
"THICKNESS (FT) : 100"
```

"LENGTH (FT) : 560"
"WIDTH (FT) : 560"
"GRID BLOCKS : 100x100x3"

COMPOSITIONAL_MODEL
$DEBUGS

TIMEEND = 730

$ I/O OPTIONS

OUTLEVEL = 1
$SPLINEOUT
$GEOMOUT
PROCOUT
OUTPUT_PRE
$OUTPUT_NPH
OUTPUT_SAT
OUTPUT_OIL
OUTPUT_GAS
$OUTPUT_DEN
OUTPUT_WEL
OUTPUT_HIS
WELLFILE = "3COMP100x100x3.WEL"

HISDATA_NUM = 100
OUTPUT_TIME() = 10 100 200 300 730
$NO_CRASH

$OUTPUT FREQUENCY
ISTEP(,,)=1
JSTEP(,,)=1
KSTEP(,,)=1

$ FAULT BLOCK AND MESH DATA
METHOD = 2
DOWN() = 0 0 1

```
NX(1) = 100  NY(1) = 100  NZ(1) = 3
MES = "cart"
DX() = 80  DY() = 80  DZ() = 20 30 50

$ COMPOUND NAMES
COMPOUND(1) = "C1"      COMPOUND(2) = "C3"
COMPOUND(3) = "C10"

$ COMPOUND CRITICAL TEMPERATURES
CRIT()  343.0 665.7 1111.8

$ COMPOUND CRITICAL PRESSURES
CRIP()  667.8 616.3  304.0

$ COMPOUND CRITICAL VOLUMES
CRIV()  1.599 3.211 10.087

$ COMPOUND ACEN
ACEN()  0.013 0.152 0.488

$ COMPOUND MOL WEIGHTS
MOLW()  16.0 44.1 142.3

$ COMPOUND PARA
PARA()  71.00 151.0 431.0

$ BINARY INTERACTION COEFFICIENTS
BINC(,) = 0.0   0.0   0.0
          0.0   0.0   0.0
          0.0   0.0   0.0

$ MAX NUMBER OF PHASES
NPHASE = 3

$ MAXNEWT MAX NUMBER OF NEWTON ITERATION
MAXNEWT = 20

$ Initial rock & water properties
```

ROCKZ = 0.000001 ROCKP = 1500
H2OZ = 0.000003 H2OP = 14.696 H2OD = 3.468
SURTF = 60.0 SURPS = 14.696
RESTF = 160.0

$ TOLERANCE
CVGOPT = 2
TOL_FLASH = 0.0001
TOL_VOLUME = 0.0001
TOL_MASS = 0.0001
TOL_WATER = 0.0001

$ POROSITY
POROSITY1() = 0.35

$ PERMEABILITIES
XPERM1() = 10
YPERM1() = 10
ZPERM1() = 10
XYPERM1() = 0
XZPERM1() = 0
YZPERM1() = 0

$ INITIAL WATER SATURATION
SWINI1() = 0.17

$ INITIAL WATER CELL PRESSURE
PINI1() = 1500.0

$ INITIAL PHASE VISCOSITIES AT EACH CELL
VIS1() = 1.0

$ INITIAL COMPOSITIONS
ZXY1(,,,1) = .5
ZXY1(,,,2) = .03
ZXY1(,,,3) = .47

$ RELPERM DATA

```
$ RELP 1 for table lookup, 2 for function based

RELP 2
$MODREL(1) = 3

$ NRELFUN 1 for corey, more to be added later
NRELFUN  1
$ data for each phase : water, phase 2 and phase 3
ENDPT() = 0.4 0.9 0.9
SR() = 0.3 0.1 0.0
EXPN() = 3.0 2.0 2.0

$ ============== WELL SPECIFICATIONS ==============

NUMWELL = 2

$ --- The first well ---
WELLNAME(1) = "INJECTOR 1"
KINDWELL(1) = 2
WELLTOP(1 TO 3,1,1) = 40 40 0
WELLBOTTOM(1 TO 3,1,1) = 40 40 100
DIAMETER(1,1) = 1.0
PRLIMIT(1) = 14695
WELLPQ(1) Block
  Interpolation Linear
  Extrapolation Constant
  Data  0.      5000.
EndBlock
$ --- The 2nd well ---
WELLNAME(2) = "PRODUCER 1"
KINDWELL(2) = 3
WELLTOP(1 TO 3,1,2) = 7960 7960 0
WELLBOTTOM(1 TO 3,1,2) = 7960 7960 100
DIAMETER(1,2) = 1.0
WELLPQ(2) Block
  Interpolation Linear
  Extrapolation Constant
  Data  0.   1300.
```

```
EndBlock
EndInitial
$ TRANSIENT DATA INPUT BLOCKS
BeginTime  0.0
TIME_CONTROL = 2
DELTIM = 1  DTIMMUL = 1.0  DTIMMAX = 30  DTIMMIN = 0.1
TUNE = 0.5  DCMAX = 0.5  DAQCMAX = 0.5  DPMAX = 0.5  DSMAX = 0.5
$MAXMOL = 1  MAXP = 10000  ERRLIMIT = 0.2
WZ() 0.77 0.20 0.03 0.0
EndTime
```

Appendix D: User's Manual of GPAS with the Partition Method, FICM, and GICM

This section will briefly present a user's guide of the new version of GPAS with Partition method, FICM, and GICM.

D.1 New Added Index

LSOLVER	SOLVER's name
0	CFIM
1	SAMG
2	BICG
10	GICM
11	FICM

D.2 Options of New Added Index

$LSOLVER:0 CFIM,1: SAMG, 2 BICG, 10 GICM, 11 FICM

```
LSOLVER = 10
```

D.3 New Input File

```
$LSOLVER:0 CFIM,1: SAMG, 2 BICG, 10 GICM, 11 FICM
LSOLVER = 10
D  4-COMPONENT"
DESCRIPTION()=
"THICKNESS (FT) : 30"
"LENGTH (FT) : 320"
"WIDTH (FT) : 320"
"GRID BLOCKS : 8x8x5"
COMPOSITIONAL_MODEL
$DEBUGS
TIMEEND = 1500
$ I/O OPTIONS
OUTLEVEL = 1
$SPLINEOUT
$GEOMOUT
PROCOUT
OUTPUT_PRE
$OUTPUT_NPH
OUTPUT_SAT
OUTPUT_OIL
OUTPUT_LIQ_2
OUTPUT_TEMP
OUTPUT_GAS
OUTPUT_DEN
OUTPUT_WEL
OUTPUT_HIS
WELLFILE = "4COMP.WEL"
OUTPUT_TIME() =  10 20 30 40 50 100 200 300 500 900
ISTEP(,,)=1
JSTEP(,,)=1
```

```
KSTEP(,,)=1
$ FAULT BLOCK AND MESH DATA
METHOD = 2
DOWN() = 0 0 1
NX(1) = 16  NY(1) = 16  NZ(1) = 3
MES = "cart"
DX() = 40  DY() = 40  DZ() = 10

$ COMPOUND NAMES
COMPOUND(1) = "C6"
COMPOUND(2) = "C17"
COMPOUND(3) = "C19"
COMPOUND(4) = "WATER"
WATERCOMP

$ COMPOUND CRITICAL TEMPERATURES
CRIT() 913.4 1319.73 1361.13 1165.47

$ COMPOUND CRITICAL PRESSURES
CRIP() 436.9  191.1 161.7 3206.2

$ COMPOUND CRITICAL VOLUMES
CRIV() 5.923 16.33 19.5 0.91465

$ COMPOUND ACEN
ACEN() 0.301 0.77 0.827 0.344


$ COMPOUND MOL WEIGHTS
MOLW() 86.2 240.475 268.529  18.015

$ COMPOUND PARA
PARA() 271.0 643.0 825.0 52.0
```

```
$ MAX NUMBER OF PHASES
NPHASE = 4

$ Linear Solver Choices
$ 0  PETSc; 1 WJPM; 2 WJAA; others PETSc;
LSOLVER = 0

$ MAXNEWT MAX NUMBER OF NEWTON ITERATION
MAXNEWT = 20

$ Initial rock & water properties
ROCKZ = 0.0001  ROCKP = 14.7
H2OZ = 0.000003  H2OP = 14.696  H2OD = 3.468
SURTF = 60.0  SURPS = 14.696
RESTF = 240.0

$ TOLERANCE
CVGOPT = 1
TOL_FLASH = 0.001
TOL_VOLUME = 0.001
TOL_MASS = 0.001
TOL_WATER = 0.001
TOL_ENERGY = 0.001
$ POROSITY
POROSITY1() = 0.3

$ PERMEABILITIES
XPERM1() = 150
YPERM1() = 150
ZPERM1() = 150
XYPERM1() = 0
XZPERM1() = 0
YZPERM1() = 0
```

```
$ INITIAL WATER SATURATION
SWINI1() = 0.000000000000017

$ INITIAL WATER CELL PRESSURE
PINI1() = 450

$ INITIAL PHASE VISCOSITIES AT EACH CELL
VIS1() = 1.0

$ INITIAL COMPOSITIONS
ZXY1(,,,1) = 0.03
ZXY1(,,,2) = 0.07
ZXY1(,,,3) = 0.15
ZXY1(,,,4) = 0.75

RELP 2
NRELFUN  1
ENDPT() = 0.5 0.7 0.8
SR() = 0.1 0.2 0.06
EXPN() = 2.5 2.0 1.5
ENERGY
ACPR1 = 35.0
HKCONST
HKROCK = 35.0
HEATLOSS
HKCONSTS
HKROCKS = 20.0
ACPI(,)= -4.4082 0.5815 -0.0003116 0.00000006504 0.0
        -13.96606155 1.62E+00 -0.000907275 1.97173E-07 0.0
         -15.48993637 1.81E+00 -0.001014407 2.2049E-07 0.0
         32.24 0.001924 0.00001055 -0.000000003596 0.0
TMEINI1() = 560.0
$ ============== WELL SPECIFICATIONS =============
NUMWELL = 2
```

```
$ --- The first well ---
WELLNAME(1) = "INJECTOR 1"
KINDWELL(1) = 2
WELLTOP(1 TO 3,1,1) = 220 220 0
WELLBOTTOM(1 TO 3,1,1) = 220 220 30
DIAMETER(1,1) = 1.0
TEMPERATURE(1) = 790.0
PRLIMIT(1) = 14695
WELLPQ(1) Block
  Interpolation Linear
  Extrapolation Constant
    Data  0.    2200.0
EndBlock

$ --- The 2nd well ---
WELLNAME(2) = "PRODUCER 1"
KINDWELL(2) = 3
WELLTOP(1 TO 3,1,2) = 620 620 0
WELLBOTTOM(1 TO 3,1,2) = 620 620 30
DIAMETER(1,2) = 1.0
WELLPQ(2) Block
  Interpolation Linear
  Extrapolation Constant
  Data  0.   400.
EndBlock
EndInitial
$ TRANSIENT DATA INPUT BLOCKS
BeginTime  0.0
TIME_CONTROL = 2
DELTIM = 0.01  DTIMMUL = 1.0  DTIMMAX = 30 DTIMMIN = 0.00001
TUNE = 0.5  DCMAX = 0.5  DAQCMAX = 0.5  DPMAX = 0.5  DSMAX = 0.5  DTMAX = 0.5
WZ() 0.0001 0.0001 0.0001 0.9997 0
EndTime
```

Bibliography

Alcouffe, R.E., Brandt, A., Dendy, J.E., and Painter, J.W. 1981. "The Multi-Grid Method for the Diffusion Equation with Strongly Discontinuous Coefficients," SIAM J. Sci. Stat. Comput. 2, pp 430-454.

Babu, D. K., Odeh, A. S., Al-Khalifa, A. J., and McCann, R. C. 1990. "The Relation Between Wellblock and Wellbore Pressures in Numerical Simulation of Horizontal Wells General Formulas for Arbitrary Well Locations in Grids," unsolicited paper SPE 20161 (1990), available from SPE, Richardson, TX.

Babu, D. K., Odeh, A. S., Al-Khalifa, A. J., and McCann, R. C. 1991. "Numerical Simulation of Horizontal Wells," Paper SPE 21425 presented at SPE Middle East Oil Show, Bahrain (Nov. 16-19, 1991).

Babu, D. K., Odeh, A. S., Al-Khalifa, A.J., and McCann, R. C. 1991. "The Relation Between Wellblock and Wellbore Pressures in Numerical Simulation of Horizontal Wells," SPE Reservoir Engineering, Aug, 1991, 324-328.

Bang, Vishal. 2005. "Phase Behavior Study of Hydrocarbon-Water-Alcohol Mixture," MS. Thesis, The University of Texas at Austin.

Bear, J. 1972. "Dynamics of Fuids in Porous Media," American Elsevier Publishing Co., Inc., NY.

Brantferger, K. M, Pope, G.A, and Sepehrnoori, K. 1991. "Development of a Thermodynamically Consistent, Fully Implicit, Compositional, Equation-Of-State, Steamflood Simulator," Paper SPE 21253 presented at the SPE Symposium on Reservoir Simulation in Anaheim, CA, February 17-20.

Brandt, A. 1986. "Algebraic Multigrid Theory: The Symmetric Case," Appl. Math. Comp. 19 pp. 23-56.

Brandt, A., McCormick, S.F., and Ruge, J. 1982. "Algebraic Multigrid (AMG) for Automatic Multigrid Solution with Application to Geodetic Computations," Institute for Computational Studies, POB 1852, Fort Collins, Colorado.

Brandt, A., McCormick, S.F., and Ruge, J. 1984. "Algebraic multigrid (AMG) for sparse matrix equations," Sparsity and its Applications, D.J. Evans (ed.), Cambridge University Press, pp 257-284, Cambridge.

Campbell, W. G. 1990. "Form and Style in Thesis Writing, a Manual of Style," Chicago: The University of Chicago Press.

CASA, 2008. “HYPRE User’s Manual, version2.4.0b,” Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, August.

Chang, Y. 1990. “Development of a Three-Dimensional, Equation-of-State Compositional Reservoir Simulator for Miscible Gas Flooding,” Ph.D. Dissertation, The University of Texas at Austin.

Chang, Y., Pope, G. A., and Sepehrnoori, K. 1990. “A Higher-Order Finite Difference Compositional Simulator,” J. Phys. Sci. and Eng., 5, Nov., 1990.

Chan, M.Y.S. and Sarioglu, G. 1992. “Numerical Modeling of Cyclically Steamed and Fractured Oil-Sand Reservoirs,” Paper SPE 22369 presented at International Meeting on Petroleum Engineering, Beijing, China, March 24-27.

Chein, M.C.H., Yardumain, H.E., Chung, E.Y., and Todd, W.W. 1989. “The formulation of a Thermal Simulation Model in a Vectorized General Purpose Reservoir Simulator,” Paper SPE presented at the 10th SPe symposium on Reservoir Simulation, Houston, TX, February.

Cicek, O. 2005. “Numerical Simulation of Steam Displacement of Oil in Naturally Fractured Reservoirs Using Fully Implicit Compositional Formulation: A Comparative Analysis of the Effects of Capillary and Gravitational Forces in Matrix/Fracture Exchange Term,” Paper SPE 97005 presented at SPE Annual Technical Conference and Exhibition, Dallas, Texas, October 9-12.

Cicek, O. and Ertekin, T. 1996. “Development and Testing of a New 3-D Field Scale Fully Implicit Multi-Phase Compositional Steam Injection Simulator,” Paper SPE 35516, April.

Dean, R.H., Gai, X. L., Stone, C.M. and Minkoff, S. E. 2006. “A Comparison of Techniques for Coupling Porous Flow and Geomechanics,” SPE 79709-PA.

Delshad, M., Pope, G.A., and Sepehrnoori, K. "A Compositional Simulator for Modeling Surfactant Enhanced Aquifer Remediation," Journal of Contaminant Hydrology, **23** (4): 303-327, August 1996.

Dendy, J.E. (Jr.). 1982. “Black box multigrid,” J. Comp. Physics 48, pp. 366-386.

Faber, V. and Manteuffel, T. 1984. "Necessary and Sufficient Conditions for the Existence of a Conjugate Gradient Method," SIAM J. Numer. Anal. 21, 315-339.

Fathi-Najafabadi, N. , Han, C., Delshad, M., and Sepehrnoori, K. 2009. “Development of Three Phase, Fully Implicit, Parallel Chemical Flooding Simulator,” Paper(SPE 119002) was presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, February.

Fazelipour, W., Pope, G. A., and Sepehrnoori, K. 2008. “Development of a Fully Implicit, Parallel, EOS Compositional Simulator,” paper (SPE 120203) presented at the SPE Reservoir Simulation Symposium, Dallas, Texas, USA, June.

Freund, R. and Nachtigal, N. "QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems," Numer. Math. 60, 315-339, 1991.

Gai, X. L. 2004. "A coupled geomechanical and reservoir flow model on parallel computers," PhD dissertation, The University of Texas at Austin, Austin, Texas, USA.

Gai, X. L., Dean, R. H., Wheeler, M. F. and Liu, R. 2003. "Coupled geomechanical and reservoir modeling on parallel computers," Paper SPE 79700 presented at the 2003 SPE Reservoir Simulation Symposium, Houston, USA, 3-5 February.

GPAS development group. 2007. "GPAS Developer's Manual, the 1st version."

Godderij, R.R., Bruining, J., and Molenaar, J. 1999. "A Fast 3D Interface Simulator for Steamdrives," Paper SPE 59269 SPE Journal, Volume 4, Number 4, December.

Han, C., Delshad, M., Sepehrnoori, K., and Pope, G.A. 2005. "A Fully Implicit, Parallel, Compositional Chemical Flooding Simulator," paper SPE 97217 presented at the 2005 SPE Technical Conference and Exhibition held in Dallas, Texas, USA, 9-12 October.

Henson, V.E. and Yang, U.M. 2002. "BoomerAMG: a parallel algebraic multigrid solver and preconditioner," Applied Numerical Mathematics, 41(5):155–177, 2002. Also available as LLNL technical report UCRL-JC-141495.

Ishimoto, K. 1985. "One dimensional Fully Implicit Compositional Model for Steam Flooding," MS Thesis, University of Texas, Austin, August.

Ishimoto, K., Pope, G.A., and Sepehrnoori, K. 1987. "An Equation of State Stream Simulator," In Situ, 11,1,1-37.

Jhaveri, B.S. and Youngren, G.K. 1988. "Three Parameter Modification of the Peng-Robinson Equation of State to Improve Volumetric Predictions," SPE Resrvoir Engineering, Aug 1988, 3, 1033-1040.

John, A., Han, C., Delshad, M., Pope, G.A., and Sepehrnoori. K. 2004. "A New Generation Chemical Flooding Simulator," the paper (SPE 89436) was first presented at the 2004 SPE/DOE symposium on Improved Oil Recovery, Tulsa, 17-21 April.

Killough, J. E. and Kossack, C. A. 1987. "Fifth Comparative Solution Project: Evaluation of Miscible Flood Simulators," SPE 16000 presented at the 9th SPE Symposium on Reservoir Simulation, San Antonio, TX, February 1-4, 1987.

Khan, S. A. 1992. "An Expert System to Aid in Compositional Simulation of Miscible Gas Flooding," PhD dissertation, The University of Texas at Austin, 1992.

Li, W. 2009. "Partitioned Method for Coupling New Model with GPAS," February, JIP report, 27 February.

Li, Yinghui. 2007. “A Fast and Robust Phase Behavior Modeling For Compositional Reservoir Simulation,” PhD dissertation, The University of Texas at Austin.

Lim, M. T. 1993. “Field-Scale Numerical Modeling of Multiple Contact Miscible Processes Using Horizontal Wells in Heterogeneous Reservoirs,” PhD dissertation, The University of Texas at Austin.

Lim, M.T. 1995. “Validation of Foam Option in UTCOMP for Modeling of CO2 Foam Displacements,” 1995 EOGRR Annual Report, Category C, Part III.

Liu, J. Delshad, M., Pope, G. A., and Sepehrnoori, K. 1992. Application of Higher-Order Flux-Limited Methods in Compositional Simulation. Trans. in Porous Media, 1992.

Lohrenz, J., Bray, B.G. and Clark, C.R. 1964. “Calculating Viscosities of Reservoir Fluids from their Compositions,” J. Pet. Tech., 16, 1964, 1171-1176.

Lu, B., Alshaalan, T.M. and Wheeler, M.F. “Iteratively Coupled Reservoir Simulation for Multiphase Flow,” The paper SPE110114 was prepared for presentation at the 2007 SPE Annual Technical Conference and Exhibition held in Anaheim, California, U.S.A., 11-14 November 2007.

Luo, S., and Barrufet, Maria. A. 2005. “Reservoir-Simulation study of Water-in-Oil Solubility Effect on Oil Recovery in the Steam Injection Process,” Paper SPE 89407 presented at SPE symposium on Improved Oil Recovery, Tulsa, Oklaohoma.

Maroongroge, V., Pope, G. A., and Sepehrnoori, K. 1991. “Modeling Tracers With UTCOMP,” 1991 CEOGRR Annual Report, Category A, 172-208.

Mifflin, R.T., Watts, J.W., and Weiser, A. 1991. “A Fully Coupled, Fully Implicit Reservoir Simulator for Thermal and other Complex Reservoir Processes,” Paper SPE 21252 presented at SPE Symposium on Reservoir Simulation, California, February.

Naimi-Tajdar, R., Han, C., Sepehrnoori, K., Arbogast, T.J., and Miller, M.A. “A Fully Implicit, Compositional, Parallel Simulator for IOR Processes in Fractured Reservoirs,” the paper (SPE 100079) was first presented at the 2006 SPE/DOE symposium on Improved Oil Recovery, Tulsa, 22-26 April.

Okuno, Ryosuke 2009. “Modeling of Multiphase Behavior for Gas Flooding Simulation,” PhD dissertation, The University of Texas at Austin.

Pan, F., and Sepehrnoori, K., and Chin, L.Y. 2009, “A New Solution Procedure for a Fully Coupled Geomechanics and Compositional Reservoir Simulator,” paper 119029 was prepared for presentation at the 2009 SPE Reservoir Simulation Symposium held in The Woodlands, Texas, USA, 2–4 February.

Peaceman, D. W. 1978. “Interpretation of Well-Block Pressures in Numerical Reservoir Simulation,” Soc. Pet. Eng. J., June 1978, 183-194; Trans., AIME, 265.

Peaceman, D.W. 1983. “Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability,” Soc. Pet. Eng. J., June, 1983, 3, 531-543.

Peaceman, D.W. 1991. “Representation of a Horizontal Well in Numerical Reservoir Simulation,” Paper SPE 21217 presented at the 11th SPE Reservoir Simulation Symposium, Anaheim, CA, Feb. 17-20, 1991.

Peng, D.Y. and Robinson, D.B. 1976. “A New Two-Constant Equation of State,” Ind. Eng. Chem. Fundam., 15, 1976, 59-64.

Perschke, D.R. 1988. “Equation of State Phase Behavior Modeling for Compositional Simulator,” Ph.D. Dissertation, The University of Texas at Austin, December 1988.

Perschke, D.R., Pope, G.A., and Sepehrnoori, K. 1989. “Comparison of Phase Behavior Algorithms using an Equation-of-state Compositional Simulator,” 1989 CEOGRR Annual Report, Category C, Section II.

Perschke, D.R., Pope, G.A. and Sepehrnoori, K. 1989. “Phase Identification During Compositional Simulation,” 1989 CEOGRR Annual Report, Category C, Section II.

Prevost, J. H. 1997. “Partitioned solution procedure for simultaneous integration of coupled-field problems,” Communications in Numerical Methods in Engineering, v13, no.4:239-247.

Reid, R.C., Prausnitz, J.M., and Sherwood, T.K. 1977. “The Properties of Gases and Liquids,” Third Edition, McGraw-Hill Book Company, New York.

Roper, M.K., Pope,G.A., and Sepehrnoori, K. 1991. “Simulation of Tertiary CO2 Injectivity,” 1991 CEOGRR Annual Report, Category A, 395-484.

Rubin, B. and Buchanan, W.L.1985. “A General Purpose Thermal Model,” Soc. Pet. Eng. J, 15, 2, 202-214.

Ruge, J.W., and Stüben, K. 1985. “Efficient Solution of Finite Difference and Finite Element Equations by Algebraic Multigrid (AMG),” Multigrid Methods for Integral and Differiential Equations (Paddon, D.J. Holstein H.; eds.), The Institute of Mathematics and its Applications Conference Series, New Series Number 3, pp. 169-212, Clarenden Press, Oxford.

Ruge, J.W., and Stüben, K. 1986. “Algebraic Multigrid (AMG),” In Multigrid Methods (McCormick, S.F., ed.), SIAM, Frontiers in Applied Mathematics, Vol. 5. Philadelphia.

Ruge, J.W. and Stüben, K. 1987. “Algebraic Multigrid (AMG),” In S. F. McCormick, editor, Multigrid Methods, volume 3 of Frontiers in Applied Mathematics, pages 73–130. SIAM, Philadelphia, PA.

Ruge, J.W. and Stüben, K. 1986. “Algebraic Multigrid (AMG),” In Multigrid Methods”(S. McCormick, ed.), Frontiers in Applied Mathematics, Vol 5, SIAM, Philadelphia.

Samier, P., and Onaisi, A. 2007. “A Practical Iterative Scheme for Coupling Geomechanics With Reservoir Simulation,” The paper SPE 107077 was accepted for presentation at the SPE Europec 2007, London, 11-14 June, and revised for publication.

Settari, A. 2000. “Reservoir simulation: A powerful tool for understanding and managing reservoir complexity,” A SPE Distinguished Lecturer Program (2000-2001), November.

Settari, A., and Mourits, F. M. 1995. “A Coupled Reservoir and Geomechanical Simulation System,” The paper SPE 29112 (50939) presented at the 1995 SPE Reservoir Simulation Symposium held in San Antonio, Texas, USA, 12-15 February.

Settari, A. and Mourits, F. M. 1994. “Coupling of Geomechanics and Reservoir Simulation models,” Proceedings of the Eighth International Conference on Computer Methods and Advances, v3:2151-2158.

Settari, A. and Walters, D. A. 1999. “Advances in Coupled Geomechanical and Reservoir Modeling with Applications to Reservoir Compaction,” Paper SPE 74142/51927 presented at the 1999 SPE Reservoir Simulation Symposium, Houston, Texas, USA, 14-17 February.

Settari, A., Kry, P.R., and Yee, C.T. 1989. ”Coupling of Fluid Flow and Soil behavior to Model Injection Into Unconsolidated Oil Sands,” Advances in Thermal Recovery, Vol. 28, No. 1,January-February.

Sonneveld, P. 1989. “CGS: A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems,” SIAM J. Sci. Statist. Comput. **10**, 36-52.

Stüben, K. and Clees, T. 2005. “SAMG USER’s Manual. Release Version, 22c-11,” Fraunhofer Institute SCAI, Schloss Birlinghoven, D-53754 St. Augustin, Germany.

Stüben, K. 2001. “An Introduction to Algebraic Multigrid,” Appendix in the book Multigrid by U. Trottenberg; C.W. Oosterlee; A. Schüller. Academic Press, pp. 413-532.

Stüben, K. 1983. “Algebraic multigrid (AMG): Experiences and comparisons,” Appl. Math. Comp. 13, pp. 419-452.

Tarahhom, F., Sepehrnoori, K. and Marcondes, F. 2009. “A Novel Approach to Integrate Dual Porosity Model and Full Permeability Tensor Representation in Fractures,” paper 119001 was prepared for presentation at the 2009 SPE Reservoir Simulation Symposium held in The Woodlands, Texas, USA, 2–4 February.

Tran, D., Nghiem, L., and Buchanan, L. 2005. “An Overview of Iterative Coupling Between Geomechanical Deformation and Reservoir Flow,” This paper was selected for presentation by an SPE/PS-CIM/CHOA Program Committee following review of information contained in a proposal submitted by the author(s).Paper SPE 93244 presented at the 2005 SPE/PS-CIM/CHOA International Thermal Operations and Heavy Oil Symposium held in Richardson, TX 75083-3836, U.S.A.

Tran, D., Nghiem, L., and Buchanan, L. 2005. “Improved iterative coupling of geomechanics with reservoir simulation,” Paper SPE 93244 presented at the 2005 SPE Reservoir Symposium held in Houston, Texas, USA, 31 January-2 February.

Tran, D., Settari, A., and Nghiem, L. 2004. “New iterative coupling between a reservoir simulator and a geomechanics module,” Paper SPE 78192 (88989) presented at the SPE/ISRM Rock Mechanics Conference, Irving, Texas, USA, 20-23 October.

Tran, D. and Settari, A. 2002. “New Iterative Coupling Between a Reservoir Simulator and a Geomechanics Module,” This paper (SPE 88989) was revised for publication from paper SPE 78192, presented at the 2002 SPE/ISRM Rock Mechanics Conference, Irving, Texas, 20–23 October.

Trangenstein, J.A. 1989. “Analysis of a Model and Sequential Numerical Method for Thermal Reservoir Simulation,” Presented at the European Conference on the Mathematics of Oil Recovery, University of Cambridge, England.

Turabian, K. L. 1987. “A Manual for Writers of Term Papers,”Theses, and Dissertations. 5th ed. Chicago: The University of Chicago Press.]

Turek, E.A., Metcalfe, R.S., Yarborough, L., and Robinson, R.L. 1984. “Phase Equilibria in CO2-Multicomponent Hydrocarbon Systems: Experimental Data and Improved Prediction Technique,” Soc. Pet. Eng. J., June 1984, 308-324.

[UTCHEM-Volume I]UTCHEM Group. 2003. “Volume I: User’s guide for UTCHEM-10, A Three –Dimensional Chemical Flooding Simulator,” July.

[UTCHEM-Volume II]UTCHEM Group. 2003. “Volume II: Technical Documentation for UTCHEM-10, A Three –Dimensional Chemical Flooding Simulator,” July.

[UTCOMP-Tech]UTCOMP Group, 2003. “Technical Documentation for UTCOMP 3.8,” May.

[UTCOMP-user]UTCOMP Group, 2003. “User's Guide for UTCOMP 3.8,” May.

van der Vorst, H. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems." SIAM J. Sci. Statist. Comput. 13, 631-644, 1992.

Varavei, A. and Sepehrnoori, K. 2009. “An EOS-Based Compositional Thermal Reservoir simulator,” Paper SPE 119154 presented at the 2009 SPE Reservoir Simulation symposium held in The Woodlands, TX, USA, 2-4 February.

Vinsome, P.K.W. 1976. “Orthomin, an Iterative Method for Solving Sparse Sets of Simultaneous Linear Equations,” This paper was prepared for the Fourth Symposium of Numerical Simulation of Reservoir performance of the Society of Petroleum Engineers of AIME to be held in Los Angeles, Calif., Feb. 19-20.

Voevodin, V. 1983. “The Problem of Non-Self-Adjoint Generalization of the Conjugate Gradient Method is Closed,” U.S.S.R. Comput. Maths. and Math. Phys. 23, 143-144.

Wang, C.H. 1994. “Polymer Option Program Implementation,” In-house Report, The University of Texas at Austin.

Wang, C.H., Lim, M.T., and Wang, P. 1995. “Summary of Recent Enhancements to UTCOMP,” EOGRR Annual Report, Category C, Part I.

Wang, P. 1996. “Implementation of Hentry's Law for Hydrocarbon Solubility in Water into the Simulator UTCOMP,” EOGRR Annual Report, Category C, Part II.

Wang, P. 1995. “Implementation of an Efficient Flash into UTCOMP,” EOGRR Annual Report, Category C, Part IV.

Wang, P., Balay, S., Kamy, S., Wheeler, J., Abate, J., Smith, B., and Pope, G. A. 1999. “A Fully Implicit Parallel EOS Compositional Simulator for Large Scale Reservoir Simulation,” paper SPE 51885 presented at the 1999 SPE SPE 109867 9 15th Reservoir Simulation Symposium, Houston, Texas, USA, 14-17 February.

Wang, P., Yotov, I., Wheeler, M. F., Arbogast, T., Dawson, C., Parashar, M., and Sepehrnoori, K. 1997. “A New Generation EOS Compositional Reservoir Simulator: Part I - Formulation and Discretization,” paper SPE 37979 presented at the SPE Reservoir Simulation Symposium, Dallas, Texas, USA, June.

Yang, U.M. 2006. “Parallel algebraic multigrid methods - high performance preconditioners,” In A. M. Bruaset and A. Tveito, editors, Numerical Solution of Partial Differential Equations on Parallel Computers, pages 209–236. Springer-Verlag, 2006. Also available as LLNL technical,report UCRL-BOOK-208032.

Vita

Wenjun Li earned Bachelor Science degree in applied mathematics from China University of Petroleum in 1988. He was an assistant professor at China University of Petroleum from 1988 to 1991. In 1993, he obtained a Master of Science degree in computational mathematics from Nanjing University in Nanjing. He also worked as a lecturer at the China University of Petroleum in Shandong from 1993 to 1998. In 2001, he earned a Doctor of Engineering in computer theory and application from Institute of Software at Chinese Academy of Sciences in Beijing. He was a postdoctoral researcher in TICAM at The University of Texas at Austin from 2001 to 2002, and a research associate at Southern Methodist University in Dallas from 2002 to 2003 and from 2005 to 2007. He was also a research fellow at The University of Texas Southwestern Medical Center in Dallas from 2003 to 2005. He worked at Precision Reservoir Modeling Inc. in Houston as a software developer and reservoir engineer for a reservoir simulator Radial-X in 2007. He enrolled in a Master of Science in petroleum engineering at The University of Texas at Austin in June, 2008.

Permanent address: Wenjun Li
266 Grove Street, Unit 10
Newton, MA 02466

This thesis was typed by Wenjun Li.